Security Target for
CypherNET
Multi-Protocol Encryptor

Compliant to the Common Criteria

Version

Version 1.18 15-Feb-08

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Overview

This document provides a complete and consistent statement of the security enforcing functions and mechanisms of the Target of Evaluation (TOE). The TOE consists of two parts, CypherNET and CypherManager.

The ST details the TOE security requirements and the countermeasures proposed to address the perceived threats to the assets protected by the TOE.

CypherNET is a high-speed, standards based multi-protocol encryptor specifically designed to secure voice, data and video information transmitted over SONET/SDH, Asynchronous Transfer Mode (ATM), Ethernet and protocol independent point-to-point data networks at data rates up to 10 Gigabits per second. It also provides access control facilities using access rules for each defined SONET/SDH, ATM, Ethernet or link connection.

CypherManager is a Graphical User Interface (GUI) software package that runs on Windows platforms. It acts as a Certification Authority (CA) for signing X.509 certificates and provides secure remote installation of X.509 certificates into CypherNET using SNMPv3.

CypherManager can also be used to securely remotely manage CypherNet encryptors. It can be used to securely set and monitor CypherNet internal configuration parameters.

1.2 Common Criteria Conformance

CypherNET together with CypherManager is Part 2 Conformant and Part 3 Conformant to the Common Criteria. The TOE is conformant to Evaluation Assurance Level EAL4.

1.3 Protection Profile Claim

The TOE has not been designed to comply with any known Protection Profile and accordingly no claim is made.

1.4 Identification

This section provides information needed to identify and control this Security Target and its Target of Evaluation.

1.4.1 Common Criteria Identification

Common Criteria for Information Technology Security Evaluation, Version 2.1, August 1999

1.4.2 Security Target Identification

ST Title: CypherNET Security Target

ST Version: 1.18

ST Issue Date: Feb 2008

1.4.3 TOE Identification

CypherNET SONET/SDH, CypherNET ATM, CypherNET LINK, and CypherNET ETHERNET models are marketing names used to describe specific derivations of CypherNET, which have restricted functionality. Any reference to CypherNET in this document also applies to these models.

CypherNet Model numbers applicable to this evaluation are listed in Table 2 and Table 3.

The CypherNet Application Software and CypherManager versions pertinent to this evaluation are as follows:

Description	Version	Applicable CypherNet Model Numbers
CypherNet Application Software	1.7.0	All units excluding: A2101B002: CYPHERNET ETHERNET 1G AC Unit A2102B002: CYPHERNET ETHERNET 1G DC Unit A2103B001: CYPHERNET 10/100 BASE-TX AC Unit A2104B001: CYPHERNET 10/100 BASE-TX DC Unit
CypherNet Application Software	1.7.5	A2101B002: CYPHERNET ETHERNET 1G AC Unit A2102B002: CYPHERNET ETHERNET 1G DC Unit A2103B001: CYPHERNET 10/100 BASE-TX AC Unit A2104B001: CYPHERNET 10/100 BASE-TX DC Unit
CypherManager	6.3.0	Applies to all units

Table 1 – CypherNET Application Software and CypherManager Versions

1.4.4 CypherNET AC Models

ID	Description
A2101B002	CYPHERNET ETHERNET 1G AC Unit
A2103B001	CYPHERNET 10/100 BASE-TX AC Unit
A2111B001	CYPHERNET ATM E1 AC Unit
A2113B001	CYPHERNET ATM E1/T1 AC Unit
A2115B001	CYPHERNET ATM E3/T3 AC Unit
A2121B001	CYPHERNET ATM OC3 Single Mode 15KM AC Unit
A2123B001	CYPHERNET ATM OC3 Single Mode 40KM AC Unit
A2125B001	CYPHERNET ATM OC3 Multi-Mode 2KM AC Unit
A2127B001	CYPHERNET ATM OC3 Single Mode 15KM & Single Mode 40KM AC Unit
A2129B001	CYPHERNET ATM OC3 Multi-Mode 2KM & Single Mode 15KM AC Unit
A2109B001	CYPHERNET ATM OC3 Multi-Mode 2KM & Single Mode 40KM AC Unit
A2117B001	CYPHERNET ATM OC12 Single Mode 15KM AC Unit
A2119B001	CYPHERNET ATM OC12 Single Mode 40KM AC Unit
A2107B001	CYPHERNET ATM OC12 Single Mode 15KM & Single Mode 40KM AC Unit
A2131B001	CYPHERNET LINK E1 AC Unit
A2133B001	CYPHERNET LINK X.21/V.11 AC Unit
A2141B001	CYPHERNET SONET OC3/OC12/OC48 AC Unit

Table 2 – CypherNET AC Model Numbers

1.4.5 CypherNET DC Models

ID	Description
A2102B002	CYPHERNET ETHERNET 1G DC Unit
A2104B001	CYPHERNET 10/100 BASE-TX DC Unit
A2112B001	CYPHERNET ATM E1 DC Unit
A2114B001	CYPHERNET ATM E1/T1 DC Unit
A2116B001	CYPHERNET ATM E3/T3 DC Unit
A2122B001	CYPHERNET ATM OC3 Single Mode 15KM DC Unit
A2124B001	CYPHERNET ATM OC3 Single Mode 40KM DC Unit
A2126B001	CYPHERNET ATM OC3 Multi-Mode 2KM DC Unit
A2128B001	CYPHERNET ATM OC3 Single Mode 15KM & Single Mode 40KM DC Unit
A2130B001	CYPHERNET ATM OC3 Multi-Mode 2KM & Single Mode 15KM DC Unit
A2110B001	CYPHERNET ATM OC3 Multi-Mode 2KM & Single Mode 40KM DC Unit
A2118B001	CYPHERNET ATM OC12 Single Mode 15KM DC Unit
A2120B001	CYPHERNET ATM OC12 Single Mode 40KM DC Unit
A2108B001	CYPHERNET ATM OC12 Single Mode 15KM & Single Mode 40KM DC Unit
A2132B001	CYPHERNET LINK E1 DC Unit
A2134B001	CYPHERNET LINK X.21/V.11 DC Unit
A2142B001	CYPHERNET SONET OC3/OC12/OC48 DC Unit
A2201B001	CYPHERNET SONET OC192 DC Unit

Table 3 – CypherNET DC Model Numbers

1.5 References

1. Common Criteria for Information Technology Security Evaluation. Version 2.1 August 1999
2. Australian Communications-Electronic Security Instruction 33 (ACSI 33)
3. ATM Security Specification Version 1.1 af-sec-0100.002 March 2001
4. FIPS PUB 46-3 Data Encryption Standard
5. FIPS PUB 81 DES Modes of Operation
6. FIPS PUB 180-1 Secure Hash Algorithm
7. FIPS PUB 186-2 Digital Signature Standard
8. FIPS PUB 197 Advanced Encryption Standard
9. NIST Special Publication SP800-38A Recommendation for Block Cipher Modes of Operation
10. PKCS #1 v2.0 RSA Cryptography Standard, RSA Laboratories July 14, 1998
11. PKCS 12 v1.0: Personal Information Exchange Syntax, RSA Laboratories June 24, 1999
12. RFC 1661 PPP Link Control Protocol, IETF, July 1994
13. RFC 1968 The PPP Encryption Control Protocol (ECP), IETF, June 1996
14. RFC 1969 The PPP DES Encryption Protocol, IETF, June 1996
15. RFC 2246 The TLS Protocol Version 1.0, IETF, January, 1999
16. RFC 2401 IPSec Security Architecture, IETF, November, 1998
17. RFC 2402 Authentication Header, IETF, November, 1998
18. RFC 2403 HMAC MD5 in ESP and AH, IETF, November, 1998
19. RFC 2404 HMAC SHA-1 in ESP and AH, IETF, November, 1998
20. RFC 2405 DES CBC Explicit IV mode, IETF, November, 1998
21. RFC 2406 Encapsulating Security Payload (ESP), IETF, November, 1998
22. RFC 2408 ISAKMP, IETF, November, 1998
23. RFC 2409 IKE Internet Key Exchange Protocol, IETF, November, 1998
24. RFC 2412 OAKLEY key determination protocol, IETF, November, 1998
25. RFC 2459 Internet X.509 Public Key Infrastructure IETF, January 1999
26. RFC 2574 User-based Security Model for version 3 of the Simple Network Management Protocol, IETF, April 1999

1.6 Glossary of Key Terms

AH	Authentication Header
ATM	Asynchronous Transfer Mode
CA	Certification Authority
CC	Common Criteria
CLP	Cell Loss Priority
CRC	Cyclic Redundancy Check
DES	Data Encryption Standard
FIPS PUB	Federal Information Processing Standard Publication
GFC	Generic Flow Control
HEC	Header Error Check
IKE	Internet Key Exchange
IP	Internet Protocol
MAC	Media Access Control
MASTER KEY	Key used to encrypt session keys
MBPS	Megabits per second
OAM	Operation and Maintenance management cells
OSP	Organisational Security Policy
PP	Protection Profile
PTI	Payload Type Indicator
PVC	Permanent Virtual Circuit
PVP	Permanent Virtual Path
RFC	Request for Comment
RSA	Public Key Algorithm
SESSION KEY	Key used to encrypt the payload of an ATM cell
SAR	Security Assurance Requirement
SFP	Security Functional Policy
SFR	Security Functional Requirement
SNMPv3	Simple Network Management Protocol Version 3
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSS	TOE Summary Specification
CAT	Virtual Channel Action Table
VC	Virtual Circuit
VP	Virtual Path
VPI/VCI	Virtual Path Identifier/Virtual Channel Identifier
X.509	Digital Certificate Standard

2 TOE Description

2.1 Overview

CypherNET is a high-speed, standards based multi-protocol encryptor specifically designed to secure voice, data and video information transmitted over Synchronous Optical/Synchronous Digital Hierarchy (SONET/SDH), Asynchronous Transfer Mode (ATM), and Ethernet Networks and protocol independent point-to-point data networks at data rates up to 10 Gigabits per second. It also provides access control facilities using access rules for each defined SONET/SDH, ATM, Ethernet, or link connection. CypherNET supports DES, TDES and AES algorithms.

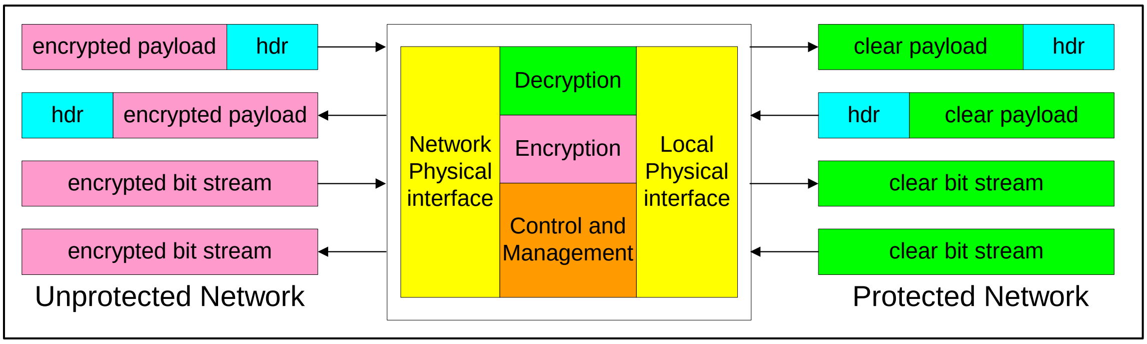


Figure 1 – CypherNET Block Diagram

Plug in interface cards enable CypherNET to be customised in the field for connection to the required network. Different interface cards can be used for the protected and unprotected ports enabling the encryptor to act as a gateway between different physical networks.

CypherNET connects to the SONET/SDH network using OC-192/STM-48, OC-48/STM-16, OC12/STM-4 or OC-3/STM-1 multimode/single mode fiber connections. Both line and path encryption modes are supported. When operating at full bandwidth, CypherNET will not discard any valid payloads for all modes of operation.

CypherNET connects to the ATM network using OC-12c/STM-4 and OC-3c/STM-1 multimode/single mode fiber, a BNC coaxial connection or RJ45 connection. When operating at full bandwidth, CypherNET will not discard any valid cells for all modes of operation.

CypherNET connects to the Local Area Network (LAN) or Wide Area Network (WAN) using 10/100/1000 BaseT RJ45 or fibre connectors. When operating at full bandwidth, CypherNET will not discard any valid Ethernet frames for all modes of operation.

CypherNET connects to the protocol independent point-to-point data network using X.21/V.11 or T1/E1 connections. When operating at full bandwidth, CypherNET will not discard bits in the received or transmitted bit stream.

CypherNET provides access control and authentication between secured sites and confidentiality of transmitted information by cryptographic mechanisms. The encryptor can be added to an existing network with complete transparency to the end user and network equipment. An example installation of the CypherNET SONET encryptor is shown in Figure 2, ATM encryptor is shown in Figure 3, Ethernet encryptor is shown in Figure 4 and protocol independent encryptor is shown in Figure 5.

CypherNET can be securely remotely managed by using CypherManager, a SNMPv3 compliant management station. Remote management sessions connect to CypherNET through the dedicated front panel Ethernet port or logically via the local or network interfaces.

CypherNET can also be managed locally through the front panel RS232 console port supporting a Command Line Interface (CLI) but this port cannot install X.509 certificates.

CypherNET supports different types of user roles with different privileges according to a set of pre-defined roles. The three defined roles are Administrator, Supervisor and Operator. Only the Administrator has unrestricted access to the security features of CypherNET. Only Administrators can activate X.509 certificates that are required for CypherNET to commence operation.

CypherNET provides an audit capability to support the effective management of the security features of the device. The audit capability records all management activity for security relevant events.

Any organisation using the CypherNET encryptor should ensure that an appropriate operational environment is maintained that satisfies those assumptions listed in section 3 of this Security Target.

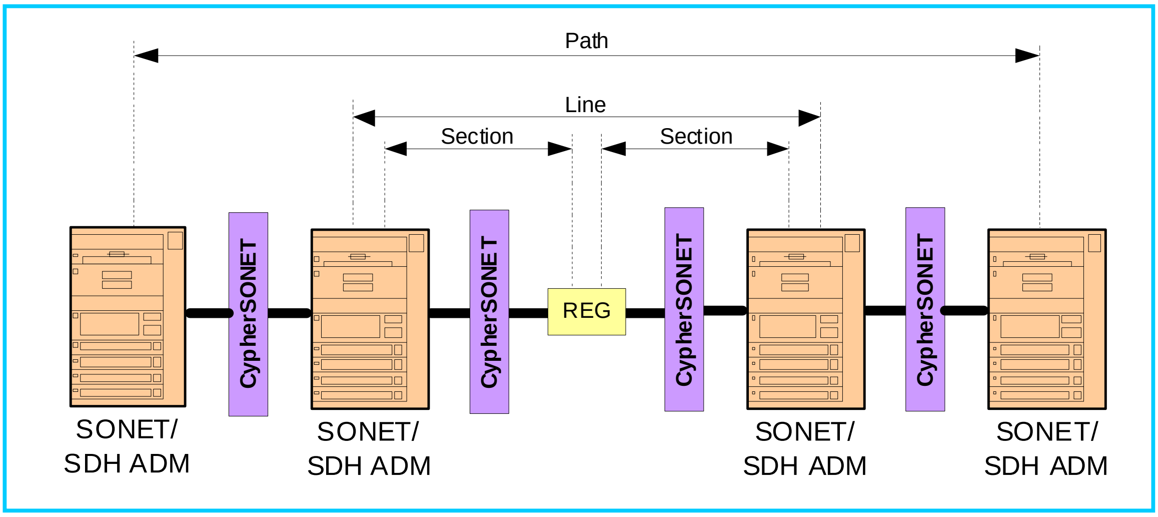


Figure 2 – SONET/SDH Security Solution

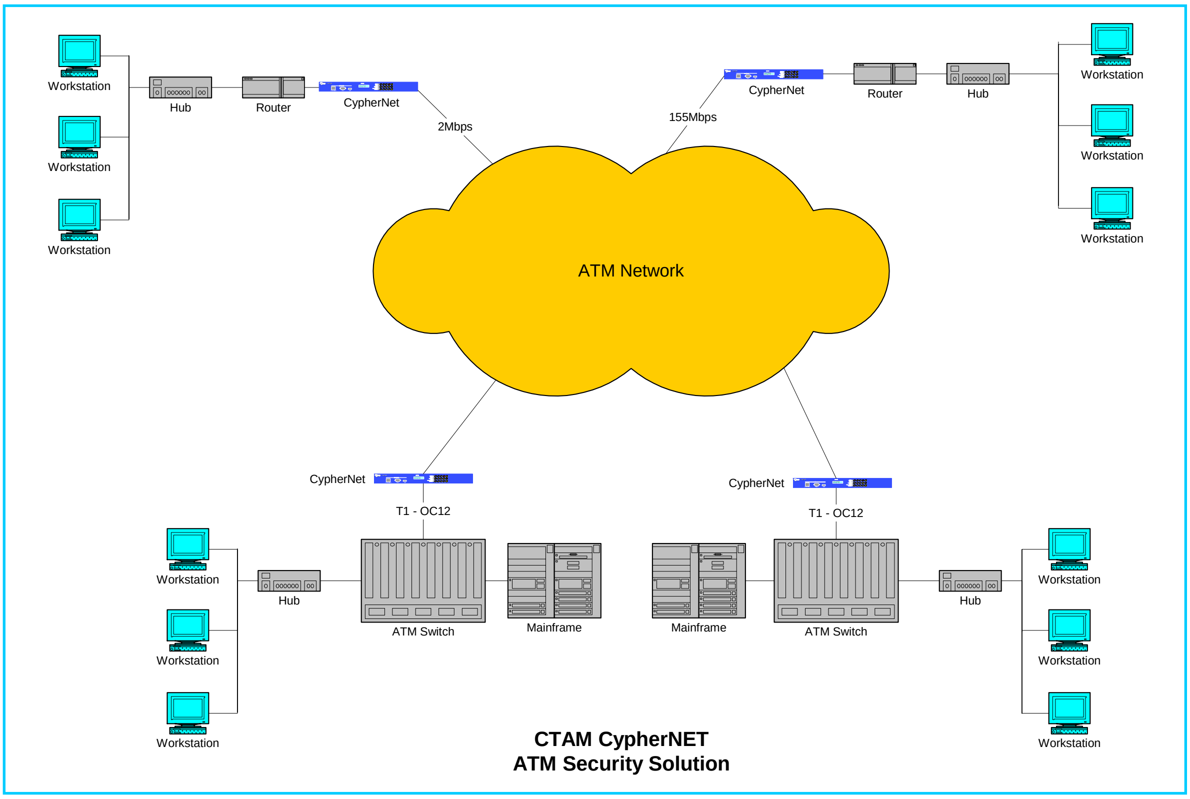


Figure 3 – ATM Security Solution

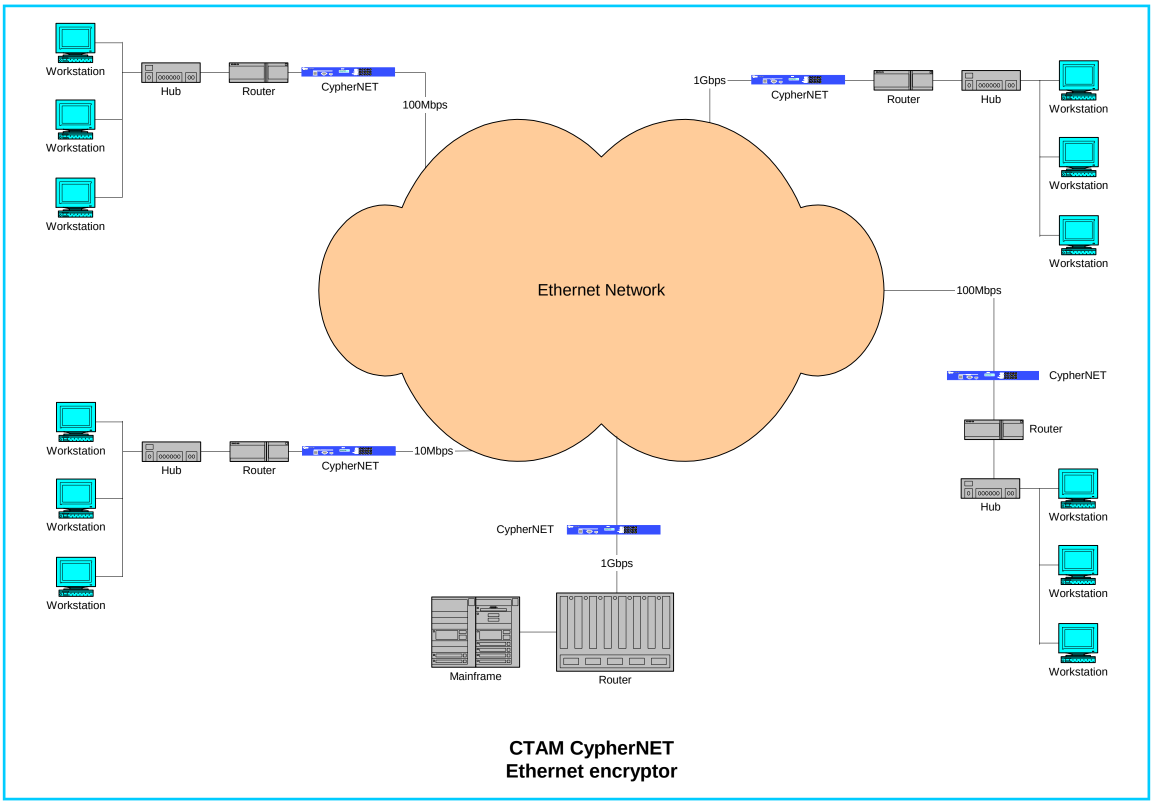


Figure 4 – Ethernet Security Solution

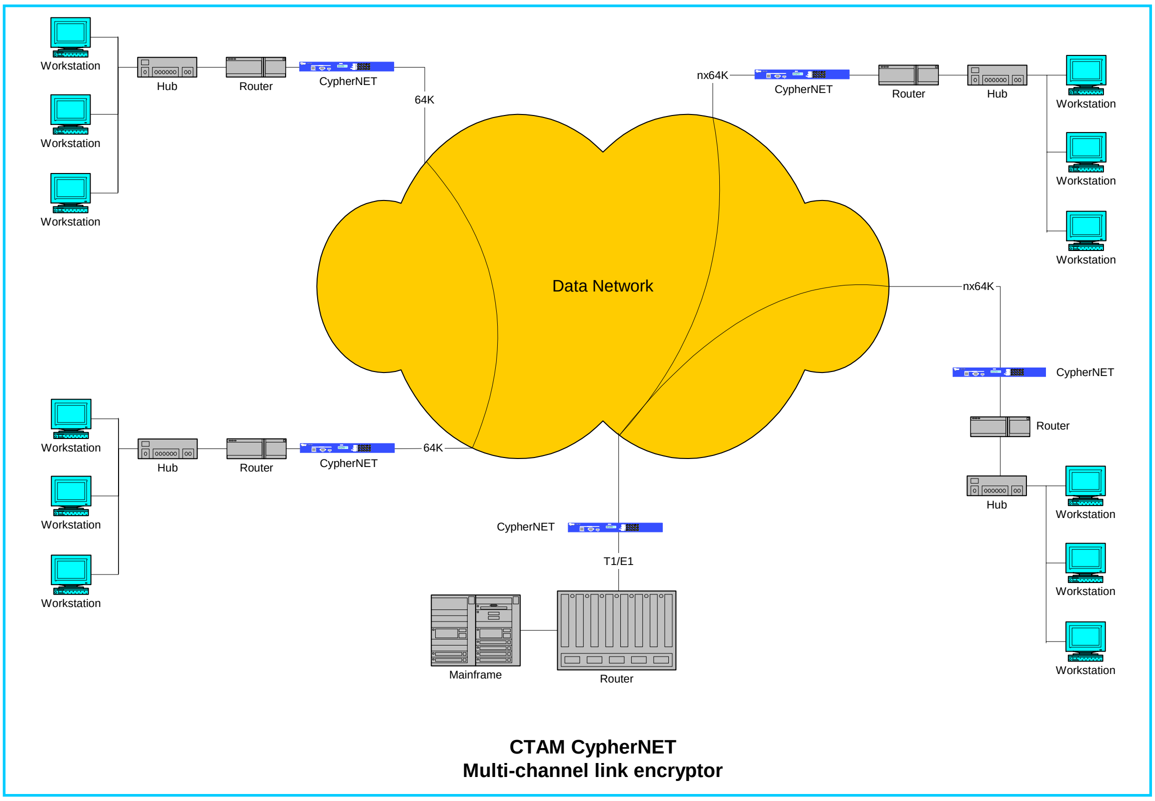


Figure 5 – Multi-Channel Link Security Solution

2.2 Security Features

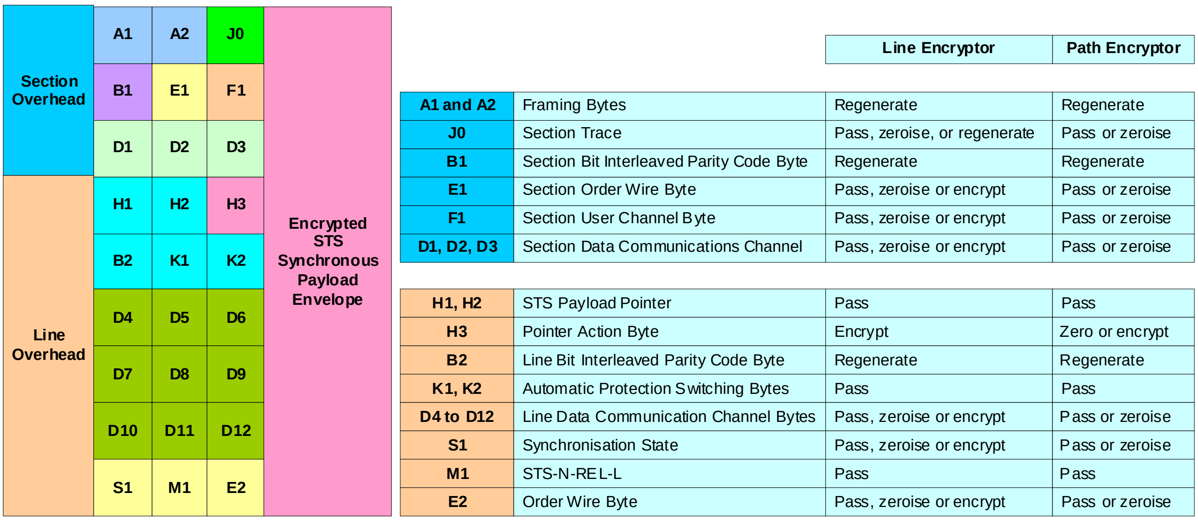
The TOE provides the following security features for each of the supported protocols.

2.2.1 SONET/SDH Processing

CypherNET provides confidentiality of the SONET/SDH frame by encrypting the payload. The line section and path overhead bytes can be individually encrypted, cleared (set to zero) or passed through unchanged depending on whether the encryptor is configured as line or path encryptor.

When configured as a line encryptor the complete STS payload, including all path overhead bytes are encrypted. When configured as a path encryptor the payload is encrypted.

The format of the SONET/SDH frame is shown in Figure 6.



		Line Encryptor	Path Encryptor
A1 and A2	Framing Bytes	Regenerate	Regenerate
J0	Section Trace	Pass, zeroise, or regenerate	Pass or zeroise
B1	Section Bit Interleaved Parity Code Byte	Regenerate	Regenerate
E1	Section Order Wire Byte	Pass, zeroise or encrypt	Pass or zeroise
F1	Section User Channel Byte	Pass, zeroise or encrypt	Pass or zeroise
D1, D2, D3	Section Data Communications Channel	Pass, zeroise or encrypt	Pass or zeroise
H1, H2	STS Payload Pointer	Pass	Pass
H3	Pointer Action Byte	Encrypt	Zero or encrypt
B2	Line Bit Interleaved Parity Code Byte	Regenerate	Regenerate
K1, K2	Automatic Protection Switching Bytes	Pass	Pass
D4 to D12	Line Data Communication Channel Bytes	Pass, zeroise or encrypt	Pass or zeroise
S1	Synchronisation State	Pass, zeroise or encrypt	Pass or zeroise
M1	STS-N-REL-L	Pass	Pass
E2	Order Wire Byte	Pass, zeroise or encrypt	Pass or zeroise

Figure 6 – SONET/SDH Frame Format

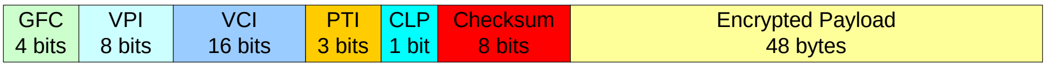
Key management and authentication use the methods described in the ATM Forum Security Specification V1.1 and uses RSA public key cryptography and X.509 certificates to provide a fully automated key management system. Master keys are transferred between encryptors using X.509 certificate authenticated RSA public key cryptography. Session keys are transferred between encryptors using master keys and are updated once every hour.

In path mode each of the available paths can be selectively encrypted. Each encrypted path uses different encryption keys for each direction.

CypherNET provides access control by discarding the payload (setting the contents to zero) if the access rule for that particular connection is violated. Access controls may be set for each connection as encrypt, bypass or discard.

2.2.2 ATM Cell Processing

CypherNET provides confidentiality of the ATM cell by encrypting the 48-byte payload and leaving the five-byte ATM header unchanged, which enables switching of the cell through ATM networks. Operation and Maintenance (OAM) cells and Virtual Path cells with Virtual Channel Identifier (VCI) values 0 to 31 can be selectively encrypted enabling ATM management functionality to be maintained. The format of a standard ATM Cell is shown in Figure 7.



GFC	VPI	VCI	PTI	CLP	Checksum	Encrypted Payload
4 bits	8 bits	16 bits	3 bits	1 bit	8 bits	48 bytes

Figure 7 – Standard ATM cell format

Key management and authentication comply with the ATM Forum Security Specification V1.1 and use RSA public key cryptography and X.509 certificates to provide a fully automated key management system. Master keys are transferred between encryptors using X.509 certificate authenticated RSA public key cryptography. Session keys are transferred between encryptors using master keys and can be set to change according to time or the number of cells encrypted.

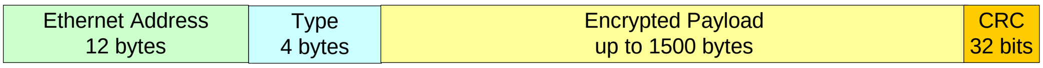
Any combination of encrypted or unencrypted virtual circuits can be configured up to a maximum of 1023 active connections for a standard ATM cell format. Each encrypted virtual circuit uses different encryption keys for each direction. Any Virtual Path Identifier/Virtual Channel Identifier (VPI/VCI) combination can be mapped to one of the 1023 available connections. Support is provided for Permanent Virtual Paths (PVP) and Permanent Virtual Circuits (PVC).

CypherNET provides access control by discarding cells if the access rules for that particular virtual circuit are violated. Access controls may be set for any VPI/VCI address as encrypt, bypass or discard

2.2.3 Ethernet Processing

CypherNET provides confidentiality of the Ethernet frame by encrypting the payload of the frame. The twelve-byte Ethernet header is unchanged, which enables switching of the frame through an Ethernet network. The format of the Ethernet frame is shown in Figure 8.

Key management and authentication comply with the ATM Forum Security Specification V1.1 and use RSA public key cryptography and X.509 certificates to provide a fully automated key management system. Master keys are transferred between encryptors using X.509 certificate authenticated RSA public key cryptography. Session keys are transferred between encryptors using master keys and can be set to change according to time or the number of frames encrypted.



Ethernet Address	Type	Encrypted Payload	CRC
12 bytes	4 bytes	up to 1500 bytes	32 bits

Figure 8 – Ethernet frame format

Any combination of encrypted or unencrypted virtual circuits can be configured up to a maximum of 1024 active connections for a standard Ethernet frame format. Each encrypted virtual circuit uses different encryption keys for each direction.

CypherNET provides access control by discarding frames if the access rules for that particular virtual circuit are violated. Access controls may be set for any Ethernet address as encrypt, bypass or discard. Ethernet management frames can be selectively encrypted or passed through in bypass enabling Ethernet management functionality to be maintained.

2.2.4 Protocol Independent Processing

CypherNET provides confidentiality of the bit stream by encrypting the bit stream using the AES or DES algorithm in single bit CFB mode.

Key management and authentication use the methods described in the ATM Forum Security Specification V1.1 and use RSA public key cryptography and X.509 certificates to provide a fully automated key management system. Master keys are transferred between encryptors using X.509 certificate authenticated RSA public key cryptography. Session keys are transferred between encryptors using master keys and can be set to change according to the time of day.

Also each of the available 24 channels in a T1 or the 32 channels in an E1 connection can be selectively encrypted. Each encrypted channel uses different encryption keys for each direction.

CypherNET provides access control by discarding the received bit stream if the access rule for that particular connection is violated. Access controls may be set for each channel as encrypt, bypass or discard.

2.3 Secure Management

The TOE provides the following secure management features.

2.3.1 Certification Authority

Each CypherNET encryptor must have an X.509 certificate, which has been signed by CypherManager acting as a Certification Authority (CA), installed before operation of the encryptor can commence.

Users of the TOE cannot use the local management RS232 port to initialise the CypherNET encryptor with an X.509 certificate. This functionality is restricted to CypherManager using an SNMPv3 management session.

2.3.2 Local Management

Local management is available via an RS232 port supporting a command line interface (CLI). Using a basic terminal emulator (not part of TOE), a user is required to present their user name and authentication password directly to the CypherNET encryptor before a local management session is allowed.

2.3.3 Remote Management using SNMPv3

CypherManager, which uses SNMPv3 management sessions, as well as acting as a CA, provides secure remote management of CypherNET. Depending on the network security policy, a user may be required to have both an authentication password and a privacy password for remote management sessions. By default, CypherManager enforces the requirement for authentication passwords, and privacy passwords are enabled at the option of the CypherNET administrator.

CypherManager, which must have IP connectivity to each CypherNET in the network, can communicate via the dedicated Ethernet management port on the front of the encryptor, which supports a 10/100BaseT connection, or via the local and network interface ports for in-band management.

3 TOE Security Environment

3.1 Assumptions

CypherNET is intended for use by organisations that need to provide confidentiality of information transmitted over SONET/SDH, ATM, Ethernet and protocol independent networks and access control to prevent unauthorised connection to the protected network. The following physical, personnel and connectivity assumptions about the operating environment and intended use of CypherNET and CypherManager apply.

Assumption	Description
Physical Assumptions	
A.CYPHERMANAGER	CypherManager is assumed to be located within controlled access facilities, which will aid in preventing unauthorised users from attempting to compromise the security functions of the TOE. For example, unauthorised physical access to the CA private key used to sign X.509 certificates. It is assumed that CypherManager will be installed on a computer with the following minimum system configuration: • Windows NT4.0/2000/XP or higher • 166MHz or higher speed processor • 64MB of memory • Hard disk drive with a minimum of 5MB of available application space • CD drive for installation • SVGA or better display resolution • Mouse or other pointing device • Network adapter card • TCP/IP connectivity
A.LOCATE	It is assumed that the CypherNET is located in a secure area at the boundary of the site to be protected. It is required to be in a secure area to ensure that the unit is not physically bypassed.
Personnel Assumptions	
A.ADMIN	It is assumed that one or more administrators, together with any other supervisors or operators, who are assigned as authorised users are competent to manage the TOE, and can be trusted not to deliberately abuse their privileges so as to undermine security.

Assumption	Description
A.AUDIT	It is assumed that appropriate audit logs are maintained and regularly examined. Without capturing security relevant events or performing regular examination of audit records, a compromise of security may go undetected.
A.PRIVATEKEY	It is assumed that a password used to protect the private key of the CypherManager remote management station is restricted to only Administrators.
Connectivity Assumptions	
A.INSTALL	It is assumed that CypherNET is installed on the boundary of the protected and unprotected network. CypherNET needs to be installed on the boundary to ensure confidentiality of transmitted information. Figure 2 shows how to secure a SONET/SDH network. Figure 3 shows how the device could be used to secure ATM networks and Figure 5 shows how to secure a protocol independent network.

Table 4 – TOE Security Environmental Assumptions

3.2 Threats

This section identifies the threats, which CypherNET is designed to counter.

The threat agents against the TOE are defined to have expertise, resources, and motivation that combine to become a low attack potential.

Threat	Description
T.ABUSE	An undetected compromise of information may occur as a result of an authorised user of the TOE (intentionally or otherwise) performing actions the individual is authorised to perform.
T.ATTACK	An undetected compromise of information may occur as a result of an attacker (insider or outsider) attempting to perform logical (i.e. non-physical) actions that the individual is not authorised to perform.
T.CAPTURE	An attacker may eavesdrop on or otherwise capture data being transmitted across a public SONET/SDH, ATM, Ethernet or data network in order to recover information that was to be kept confidential.
T.CONNECT	An attacker (insider or outsider) may attempt to make unauthorised connections to another SONET/SDH, ATM, Ethernet or data network and transmit information that was to be kept confidential, to another destination.

Threat	Description
T.IMPERSON	An attacker (outsider or insider) may impersonate an authorised user of the TOE to gain access to information that was to be kept confidential.
T.LINK	An attacker may be able to observe multiple uses of services by an entity and, by linking these uses, be able to deduce information, which the entity wishes to be kept confidential.
T.MAL	Data being transmitted across a public SONET/SDH, ATM, Ethernet, or data network may be modified or disclosed to an unauthorised individual or user of the TOE through malfunction of the TOE.
T.OBSERVE	An attacker could observe the legitimate use of the remote management service by an authorised user when that authorised user wishes their use of that remote management service to be kept confidential.
T.PHYSICAL	Security critical parts of the TOE may be subject to physical attack by an (outside or inside) attacker, which may compromise security.
T.PRIVILEGE	A compromise of information may occur as a result of actions taken by careless, willfully negligent or hostile administrators or other authorised users.

Table 5 – TOE Security Environmental Threats

3.3 Organisational Security Policies

Policy	Description
P.CRYPTO	All encryption services including, confidentiality, authentication, key generation and key management, must conform to standards specified in FIPS PUB 140-2 and ACSI33.
P.INFOFLOW	Traffic flow is controlled on the basis of the information in the SONET/SDH header, ATM cell header or Ethernet frame and the action specified in the Connection Action Table. Any SONET/SDH payloads, ATM cells, Ethernet frame or bit stream for which there is no CAT entry, are discarded. By default, all SONET/SDH payloads, ATM cells, Ethernet frames or bit streams are discarded. The P.INFOFLOW OSP ensures that the correct protective action of bypass, discard or encrypt is applied to any given SONET/SDH payload, ATM cell, Ethernet frame or bit stream received by the TOE.
P.ROLES	Administration of the TOE is controlled through the definition of roles, which assign different privilege levels to different types of authorised users (administrators, supervisors and operators). The P.ROLES OSP ensures that administration of the TOE is performed in accordance with the concept of *least privilege*.

Table 6 – TOE Security Environment Organisational Security Policies

4 Security Objectives

4.1 TOE Security Objectives

Objective	Description
O.ADMIN	The TOE must provide functionality, which enables an authorised user to effectively manage the TOE and its security functions, and must ensure that only authorised users are able to access such functionality, while also maintaining confidentiality of sensitive management data.
O.AUDIT	The TOE must provide a means to record a readable audit trail of security relevant events with accurate dates and times so as to assist in the detection of potential attacks of the TOE and also to hold users accountable for any actions that they perform.
O.CERTGEN	The TOE must provide the means for generating, issuing and managing signed X.509 certificates that conform to standards specified in FIPS PUB 140-2 and ACSI33.
O.ENCRYPT	The TOE must provide the means of protecting the confidentiality of information transferred across a public network between two protected networks using cryptography that conforms to standards specified in FIPS PUB 140-2 and ACSI33.
O.FAILSAFE	In the event of an error occurring, the TOE will preserve a secure state.
O.INFOFLOW	The TOE must provide authorised users with the means of controlling traffic flow received and transmitted on the local and network interfaces, on the basis of overhead bytes, header or channel information, in accordance with the set of rules defined in the P.INFOFLOW security policy, which includes bypass, discard or encrypt.
O.IDENT	The TOE must uniquely identify all users and authenticate the claimed identity before granting a user access to the TOE management facilities.

O.KEYMAN	The TOE must provide the means for exchanging keys with only another authorised TOE or a remote trusted IT product so the key exchange conforms to standards specified in FIPS PUB 140-2 and ACSI33.
O.ROLES	The TOE must prevent users from gaining access to and performing operations, on its resources for which their role is not explicitly authorised.
O.TAMPER	The TOE must protect itself and cryptography-related IT assets from unauthorised physical access, modification or use.
O.REMOTEMGT	The TOE must allow secure remote management of the TOE using cryptographic measures that conforms to standards specified in FIPS PUB 140-2 and ACSI33.

Table 7 – TOE Security Objectives

4.2 Environmental Security Objectives

Objective	Description
O.AUDITLOG	Authorised users of the TOE must ensure that audit facilities are used and managed effectively. In particular: a. Appropriate action must be taken to ensure that continued audit logging, e.g. by regular archiving of logs. b. Audit logs should be inspected on a regular basis, and appropriate action should be taken on the detection of breaches of security, or events that are likely to lead to a breach in the future.
O.AUTHDATA	Those responsible for the management of the TOE must ensure that the authentication data for each account on the TOE is held securely and not disclosed to persons unauthorised to use that account.
O.CONNECT	Those responsible for the TOE must ensure that no connections are provided to outside systems or users that would undermine IT security.
O.INSTALL	Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner, which maintains IT security.
O.PERSONNEL	Those responsible for the TOE are competent to manage the TOE and can be trusted not to deliberately abuse their privileges so as to undermine security.
O.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE that are critical to security policy enforcement are protected from physical attack, which might compromise IT security.
O.ROLEMGT	The administrator responsible for controlling who has access to the unit for configuration and monitoring activities must allocate users roles with the concept of *least privilege*. There are three roles: Administrator: who has full access rights; Supervisor: who has full access rights except they cannot add, delete or modify user accounts, they cannot install X.509 certificates and they cannot upgrade the firmware; and

	Operator: who can view all available information but cannot delete, add or modify the information

Table 8 – Environmental Security Objectives

5 IT Security Requirements

5.1 TOE Security Functional Requirements

A summary of functional requirements for the TOE is listed in the following table.

No.	Component	Component Name
Class FAU: Audit		
1	FAU_GEN.1	Audit data generation
2	FAU_SAR.1	Audit review
Class FCS: Cryptographic Support		
3	FCS_CKM.1	Cryptographic key generation
4	FCS_CKM.2	Cryptographic key distribution
5	FCS_CKM.4	Cryptographic key destruction
6	FCS_COP.1	Cryptographic operation
Class FDP: User Data Protection		
7	FDP_ACC.1	Subset access control
8	FDP_ACF.1	Security attribute based access control
9	FDP_DAU.1	Basic data authentication
10	FDP_IFC.1	Subset information flow control
11	FDP_IFF.1	Simple security attributes
12	FDP_UCT.1	Basic data exchange confidentiality
Class FIA: Identification and Authentication		
13	FIA_AFL.1	Authentication failure handling
14	FIA_UAU.2	User authentication before any action
15	FIA_UID.2	User identification before any action
Class FMT: Security Management		
16	FMT_MSA.1	Management of security attributes
17	FMT_MSA.2	Secure security attributes
18	FMT_MSA.3	Static attribute initialisation
19	FMT_MTD.1	Management of TSF data
20	FMT_SMR.1	Security roles
Class FPT: Protection of the TSF		

21	FPT_AMT.1	Abstract machine testing
22	FPT_FLS.1	Failure with preservation of secure state
23	FPT_ITT.1	Basic internal TSF data transfer protection
24	FPT_PHP.3	Resistance to physical attack
25	FPT_STM.1	Reliable time stamps
26	FPT_TST.1	TSF testing
Class FTA: TOE Access		
27	FTA_SSL.3	TSF-initiated termination
Class FTP: Trusted Path/Channels		
28	FTP_ITC.1	Inter-TSF trusted channel

Table 9 – TOE Security Functional Requirements

5.2 TOE Security Functions

The following sections contain the functional components from the Common Criteria Part 2 with the operations completed. The standard Common Criteria text is in regular font; the text inserted is in red italic font.

5.2.1 Security Audit (FAU)

5.2.1.1 FAU_GEN.1 – Audit data generation

Hierarchical to: No other components

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 *minimum* level of audit and

c) *FMT_MTD.1* *All modifications to the values of the TSF data*

FPT_AMT.1 *Execution of the tests of the underlying machine and the results of the tests*

FPT_FLS.1 *Failure of the TSF.*

FPT_TST.1 *Execution of the TSF self tests and the results of the tests*

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 ST,

FCS_CKM.1 *Success and failure of the activity*

FCS_CKM.2 *Success and failure of the activity*

FCS_CKM.4 *Success and failure of the activity*

FCS_COP.1 *Success and failure, and the type of cryptographic operation*

FDP_ACF.1 *Successful requests to perform an operation on an object covered by the SFP*

FDP_DAU.1 *Successful generation of validity evidence*

FDP_IFF.1 *Decisions to permit requested information flows.*

FDP_UCT.1 *The identity of any user or subject using the data exchange mechanism*

FIA_AFL.1 *The reaching of the threshold for the unsuccessful authentication attempts and the actions taken and the subsequent, if appropriate, restoration to the normal state.*

FIA_UAU.2 *Unsuccessful use of the user authentication mechanism*

FIA_UID.2 *Unsuccessful use of the user identification mechanism,*

	including the user identity provided
FMT_MSA.2	*All offered and rejected values for a security attribute*
FMT_SMR.1	*Modifications to the group of users that are part of a role*
FPT_STM.1	*Changes to the time*
FTA_SSL.3	*Termination of an interactive session by the session locking mechanism*
FTP_ITC.1	*Failure of the trusted channel functions* *Identification of the initiator and target of failed trusted channel functions*

Dependencies: FPT_STM.1 Reliable time stamps

5.2.1.2 FAU_SAR.1 – Audit review

Hierarchical to: No other components

FAU_SAR.1.1 The TSF shall provide *all authorised users* 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.

Dependencies: FAU_GEN.1 Audit data generation

5.2.2 Cryptographic Support (FCS)

5.2.2.1 FCS_CKM.1.A – Cryptographic key generation

Hierarchical to: No other components

FCS_CKM.1.1.A The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm, *DES, AES, RSA* and specified cryptographic key sizes *DES – 56 bits, 112 bits, 168 bits, AES – 128 bits, 256 bits, RSA – 1024 bits* that meet the following: *FIPS PUB 186-2 Digital Signature Standard, Appendix 3.*

Dependencies: FCS_COP.1 Cryptographic operation
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

5.2.2.2 FCS_CKM.1.B – Cryptographic key generation

Hierarchical to: No other components

FCS_CKM.1.1.B The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm, *SNMPv3 key generation using user passwords*, and specified cryptographic key sizes *SNMPv3 privacy and authentication keys – 128 bits* that meet the following: *RFC2574.*

Dependencies: FCS_COP.1 Cryptographic operation
FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

5.2.2.3 FCS_CKM.1.C – Cryptographic key generation

Hierarchical to: No other components

FCS_CKM.1.1.C The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm, *RSA, DES,* and specified cryptographic key sizes *RSA – 1024 bits, DES – 56 bits, 112 bits 168 bits*, that meet the following: *FIPS PUB 186-2 Digital Signature Standard, Appendix 3*.

Dependencies: FCS_COP.1 Cryptographic operation

FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

5.2.2.4 FCS_CKM.2.A – Cryptographic key distribution

Hierarchical to: No other components

FCS_CKM.2.1.A The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method, *RSA public key and Master/Session key using X.509 certificates for authentication*, that meets the following: *ATM Forum Security Specification V1.1, PKCS #1*

Dependencies: FCS_CKM.1 Cryptographic operation

FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

5.2.2.5 FCS_CKM.4 – Cryptographic key destruction

Hierarchical to: No other components

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method: *The keys used to encrypt the payload of the SONET/SDH frame, ATM cell, Ethernet frame, or bit stream are held in volatile memory. Loss of electrical power will destroy all session keys. If the case is opened master keys used to encrypt the RSA private key and user passwords are automatically erased* that meets the following: *none*.

Dependencies: FCS_CKM.1 Cryptographic key generation

FMT_MSA.2 Secure security attributes

5.2.2.6 FCS_COP.1.A – Cryptographic operation

Hierarchical to: No other components

FCS_COP.1.1.A The TSF shall perform *64 bit Cipher Feedback, 8 bit Cipher Feedback, 1 bit Cipher Feedback and counter mode* in accordance with a specified cryptographic algorithm, *DES* and cryptographic key sizes *56 bits, 112 bits and 168 bits* that meet the following: *FIPS PUB 46-3, FIPS PUB 81 and ATM Forum Security Specification V1.1*.

Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

5.2.2.7 FCS_COP.1.B – Cryptographic operation

Hierarchical to: No other components

FCS_COP.1.1.B The TSF shall perform *128 bit Cipher Feedback, 8 bit Cipher Feedback, 1 bit Cipher Feedback and counter mode* in accordance with a specified cryptographic algorithm, *AES* and cryptographic key sizes *128 bits and 256 bits* that meet the following: *FIPS PUB 197 and FIPS PUB SP800-38A*.

Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

5.2.2.8 FCS_COP.1.C – Cryptographic operation

Hierarchical to: No other components

FCS_COP.1.1.C The TSF shall perform *public key encryption* in accordance with a specified cryptographic algorithm *RSA* and cryptographic key sizes *1024 bits* that meet the following: *ATM Forum Security Specification V1.1, PKCS#1*.

Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

5.2.2.9 FCS_COP.1.F – Cryptographic operation

Hierarchical to: No other components

FCS_COP.1.1.F The TSF shall perform *message digest generation/verification* in accordance with a specified cryptographic algorithm *SHA-1* and cryptographic key sizes *160 bits*, that meet the following: *FIPS PUB 180-1.*

Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

5.2.2.10 FCS_COP.1.G – Cryptographic operation

Hierarchical to: No other components

FCS_COP.1.1.G The TSF shall perform *digital signature generation* in accordance with a specified cryptographic algorithm *RSA* and cryptographic key sizes *1024 bits* that meet the following: *PKCS#1*.

Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

5.2.3 User Data Protection (FDP)

5.2.3.1 FDP_ACC.1– Subset access control

Hierarchical to: No other components

FDP_ACC.1.1 The TSF shall enforce the *Management Access Control SFP* on

Subjects: Management packets, consisting of:

- *all SNMPv3 packets received on the CypherNET Ethernet management port interface and the local and network interfaces; and*
- *all data received on the CypherNET console management port interface*

Objects: Encryptor information, consisting of:

- *Channel Action Table;*
- *User Table;*
- *System Time;*
- *Audit Log;*
- *X.509 Certificate; and*
- *Firmware.*

Operations: Management operations, consisting of:

- *Viewing Channel Action Table, User Table, System Time and Audit Log;*
- *Modifying Channel Action Table, User Table and System Time;*
- *Clearing the Audit Log;*
- *Activating X.509 Certificate;*
- *Backup and restore encryptor configuration data; and*
- *Upgrading Firmware.*

Dependencies: FDP_ACF.1 Security attribute based access control

5.2.3.2 FDP_ACF.1 – Security attribute based access control

Hierarchical to: No other components

FDP_ACF.1.1 The TSF shall enforce the *Management Access Control SFP* to objects based on the

- *user's ID and the user's authentication password contained in management packets; and*
- *privacy password used to encrypt SMNPv3 management packets.*

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

- *If the User ID received on the console port interface is listed in the User Table and the authentication password in the management packet is the same as the local authentication password then console mode logon is allowed. This logon mode will allow management packets to perform the* *management operations upon the objects allowed by the user's defined role.*

- *Management packets received via SNMPv3 can also be encrypted with the privacy password associated with the User ID.*
- *If the User ID field in the encrypted SNMPv3 packet is listed in the User Table and the authentication password in the management packet is the same as the local authentication password then the management operation is allowed subject to the 'users defined role.*

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

- *none.*

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the *following rules*:

- *If the user ID received on the console port interface is not listed in the user table.*
- *If the user ID received on the console port is listed in the user table and the authentication password in the management packet is not the same as the local authentication password.*
- *If the user ID field of the SNMPv3 packet is not listed in the user table.*
- *If the user ID field of the SNMPv3 packet is listed in the user table and the data cannot be decrypted*
- *If the user ID field of the SNMPv3 packet is listed in the user table and the data can be decrypted, but the authentication check fails.*

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

5.2.3.3 FDP_DAU.1 – Basic data authentication

Hierarchical to: No other components

FDP_DAU.1.1 The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of *X.509 Certificate generation requests from a CypherNET and New X.509 Certificates generated by CypherManager for a CypherNET.*

FDP_DAU.1.2 The TSF shall provide *administrators* with the ability to verify evidence of the validity of the indicated information.

Dependencies: No dependencies

5.2.3.4 FDP_IFC.1 – Subset information flow control

Hierarchical to: No other components

FDP_IFC.1.1 The TSF shall enforce the *Information Flow Control SFP* on

Subjects: External and internal hosts which send and receive information *through the TOE*

Information: *SONET/SDH frames, ATM cells, Ethernet frames and bit streams received on the local and network interfaces*

Operation: *Encrypt, bypass or discard the received SONET/SDH frame, ATM cell, Ethernet frame or bit stream*

Dependencies: FDP_IFF.1 Simple security attributes

5.2.3.5 FDP_IFF.1 – Simple security attributes

Hierarchical to: No other components

FDP_IFF.1.1 The TSF shall enforce the *Information Flow Control SFP* based on the following types of subject and information security attributes:

- *H1 and H2 pointer bytes in the SONET/SDH frame overhead bytes*
- *VPI/VCI address contained in the ATM cell header*
- *MAC address contained in the Ethernet frame header*
- *Channel number for a bit stream connection*

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

Subjects on an internal or external network can cause information to flow through the TOE on the local and network interfaces if:

- *The Line/path number in a SONET/SDH connection, the VPI/VCI address in the ATM cell header, the MAC address in the Ethernet header or channel number in a bit stream connection is listed in the CAT then the defined operation in the CAT is allowed..*

FDP_IFF.1.3 The TSF shall enforce the *additional information flow control SFP rules:*

- *If the operation in the CAT is defined as "encrypt" then the SONET/SDH frame, ATM cell, Ethernet frame or bit stream will be passed with the SONET/SDH payload, ATM cell, Ethernet payload or bit stream encrypted/decrypted.*
- *If the operation in the CAT is defined as "bypass" then the SONET/SDH frame, ATM cell, Ethernet frame or bit stream will be passed without modification.*
- *If the operation in the CAT is defined as "discard" then the SONET/SDH frame, ATM cell, Ethernet frame or bit stream will be discarded without further action.*

FDP_IFF.1.4 The TSF shall provide the following *no additional information flow control SFP capabilities.*

FDP_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules:

- *none*

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules:

- *none.*

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialisation

5.2.3.6 FDP_UCT.1 – Basic data exchange confidentiality

Hierarchical to: No other components

FDP_UCT.1.1 The TSF shall enforce the *Information Control SFP* to be able to *transmit, receive* objects in a manner protected from unauthorised disclosure.

Dependencies: FTP_ITC.1 Inter-TSF trusted channel

FDP_IFC.1 Subset information flow control

5.2.4 Identification and Authentication (FIA)

5.2.4.1 FIA_AFL.1 – Authentication failure handling

Hierarchical to: No other components.

FIA_AFL.1.1 The TSF shall detect when *three* unsuccessful authentication attempts occur related to *the last successful authentication of a user using the console port.*

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall *disable the user account for three minutes.*

Dependencies: FIA_UAU.1 Timing of authentication

5.2.4.2 FIA_UAU.2 – User authentication before any action

Hierarchical to: FAI_UAU.1

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.

Dependencies: FIA_UID.1 Timing of identification

5.2.4.3 FIA_UID.2 – User identification before any action

Hierarchical to: FIA_UID.1

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

Dependencies: No dependencies

5.2.5 Security Management (FMT)

5.2.5.1 FMT_MSA.1.A – Management of security attributes

Hierarchical to: No other components

FMT_MSA.1.1.A The TSF shall enforce the *Information Flow Control SFP* to restrict the ability to *change default, modify* the security attributes *for each kind of information flow type:*

- *H1 and H2 pointer bytes for SONET/SDH information flows*
- *VPI/VCI address for ATM information flows*

- *MAC address for Ethernet information flows*
- *Channel number for bit stream information flows*

And the action applied to the information flow:

- *encrypt, bypass, or discard*

is listed in the CAT table to *administrators and supervisors*.

Dependencies: FDP_IFC.1 Subset information flow control

FMT_SMR.1 Security roles

5.2.5.2 FMT_MSA.1.B – Management of security attributes

Hierarchical to: No other components

FMT_MSA.1.1.B The TSF shall enforce the *Management Access Control SFP* to restrict the ability to:

- *add, delete, or modify* the security attributes *user accounts* to *administrators*
- *activate* the security attributes X*.509 certificates* to *administrators.*
- *remotely upgrade* the security attributes *firmware* to *administrators*

Dependencies: FDP_ACC.1 Subset access control

FMT_SMR.1 Security roles

5.2.5.3 FMT_MSA.2.A – Secure security attributes

Hierarchical to: No other components

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

Dependencies: ADV_SPM.1 Informal TOE security policy model

FDP_IFC.1 Subset information flow control

FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

5.2.5.4 FMT_MSA.2.B – Secure security attributes

Hierarchical to: No other components

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

Dependencies: ADV_SPM.1 Informal TOE security policy model

FDP_ACC.1 Subset access control

FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

5.2.5.5 FMT_MSA.2.C – Secure security attributes

Hierarchical to: No other components

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

Dependencies: ADV_SPM.1 Informal TOE security policy model

FDP_ACC.1 Subset access control

FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

5.2.5.6 FMT_MSA.3.A – Static attribute initialisation

Hierarchical to: No other components

FMT_MSA.3.1.A The TSF shall enforce the *Information Access Control SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

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

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

5.2.5.7 FMT_MSA.3.B – Static attribute initialisation

FMT_MSA.3.1.B The TSF shall enforce the *Management Access SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

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

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

5.2.5.8 FMT_MTD.1 – Management of TSF data

Hierarchical to: No other components

FMT_MTD.1.1 The TSF shall restrict the ability to

- *change_default, query, modify, delete and clear* the *CAT table, User Account table, X.509 certificate* to *administrators*
- *change_default, query, modify, delete and clear* the *CAT table and query the User Account table* to *supervisors.*
- *query* the *CAT and User Account tables* to *operators and above*
- *clear* the *audit log* to *administrators*
- *set* the *system time* to *administrators* and *supervisors*
- *backup and restore* the *encryptor configuration data* to *administrators and supervisors*

Dependencies: FMT_SMR.1 Security roles

5.2.5.9 FMT_SMR.1 – Security roles

Hierarchical to: No other components

FMT_SMR.1.1 The TSF shall maintain the roles *administrator, supervisor and operator*.

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

Dependencies: FIA_UID.1 Timing of identification

5.2.6 Protection of the TSF (FPT)

5.2.6.1 FPT_AMT.1 – Abstract machine testing

Hierarchical to: No other components

FPT_AMT.1.1 The TSF shall run a suite of tests *during initial start-up* to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

Dependencies: No dependencies

5.2.6.2 FPT_FLS.1 – Failure with preservation of secure state

Hierarchical to: No other components.

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur:

- *self tests return a fail result*

Dependencies: ADV_SPM.1 Informal TOE security policy model

5.2.6.3 FPT_ITT.1 – Basic internal TSF data transfer protection

Hierarchical to: No other components

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

Dependencies: No dependencies

5.2.6.4 FPT_PHP.3.A – Resistance to physical attack

Hierarchical to: No other components

FPT_PHP.3.1.A The TSF shall resist *attempts, by opening the unit, to gain physical access* to the *key material* by responding automatically such that the TSP is not violated.

Dependencies: No dependencies

5.2.6.5 FPT_PHP.3.B – Resistance to physical attack

Hierarchical to: No other components

FPT_PHP.3.1.B The TSF shall resist *attempts, by opening the unit, to gain physical access* to the *password data* by responding automatically such that the TSP is not violated.

Dependencies: No dependencies

5.2.6.6 FPT_STM.1 – Reliable time stamps

Hierarchical to: No other components

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

Dependencies: No dependencies

5.2.6.7 FPT_TST.1 – TSF testing

Hierarchical to: No other components.

FPT_TST.1.1 The TSF shall run a suite of self-tests *during initial start-up* to demonstrate the correct operation of the TSF.

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of TSF data.

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

Dependencies: FPT_AMT.1 Abstract machine testing

5.2.7 TOE Access (FTA)

5.2.7.1 FTA_SSL.3 – TSF-initiated termination

Hierarchical to: No other components.

FTA_SSL.3.1 The TSF shall terminate an interactive session after a *period of 10 minutes*.

Dependencies: No dependencies

5.2.8 Trusted Path/Channels (FTP)

5.2.8.1 FTP_ITC.1 – Inter-TSF trusted channel

Hierarchical to: No other components

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

FTP_ITC.1.2 The TSF shall permit *the TSF or the remote trusted IT product* to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for *all SONET/SDH frames, ATM cells, Ethernet frames and bit streams as defined by the Information Flow Control SFP.*

Dependencies: No dependencies

5.3 TOE Security Assurance Requirements

CypherNET together with CypherManager are intended to meet the Common Criteria EAL4 evaluation level. A summary of assurance requirements for the TOE is listed in the following table.

No.	Component	Component Name
Class ACM: Configuration management		
1	ACM_AUT.1	Partial CM automation
2	ACM_CAP.4	Generation support and acceptance procedures
3	ACM_SCP.2	Problem tracking CM coverage
Class ADO: Delivery and operation		
4	ADO_DEL.2	Detection of Modification
5	ADO_IGS.1	Installation, generation and start-up procedures
Class ADV: Development		
6	ADV_FSP.2	Fully defined external interfaces
7	ADV_HLD.2	Security enforcing high-level design
8	ADV_IMP.1	Subset of the implementation of the TSF
9	ADV_LLD.1	Descriptive low-level design
10	ADV_RCR.1	Informal correspondence demonstration
11	ADV_SPM.1	Informal TOE security policy model
Class AGD: Guidance documents		
12	AGD_ADM.1	Administrator guidance
13	AGD_USR.1	User guidance
Class ALC: Life cycle support		
14	ALC_DVS.1	Identification of security measures
15	ALC_LCD.1	Developer defined life-cycle model
16	ALC_TAT.1	Well-defined development tools
Class ATE: Tests		
17	ATE_COV.2	Analysis of coverage
18	ATE_DPT.1	Testing: high-level design
19	ATE_FUN.1	Functional testing
20	ATE_IND.2	Independent testing - sample
Class AVA: Vulnerability assessment		

21	AVA_MSU.2	Validation of analysis
22	AVA_SOF.1	Strength of TOE security function evaluation
23	AVA_VLA.2	Independent vulnerability analysis

Table 10 – TOE Security Assurance Requirements

5.4 TOE Assurance Requirements

The following sections contain the assurance requirements from the Common Criteria Part 3 for an EAL4 evaluation.

5.4.1 Configuration Management (ACM)

5.4.1.1 ACM_AUT.1 – Partial CM automation

ACM_AUT.1.1D The developer shall use a CM system.

ACM_AUT.1.2D The developer shall provide a CM plan.

ACM_AUT.1.1C The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation.

ACM_AUT.1.2C The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.1.3C The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.1.4C The CM plan shall describe how the automated tools are used in the CM system.

5.4.1.2 ACM_CAP.4 – Generation support and acceptance procedures

ACM_CAP.4.1D The developer shall provide a reference for the TOE.

ACM_CAP.4.2D The developer shall use a CM system.

ACM_CAP.4.3D The developer shall provide CM documentation.

ACM_CAP.4.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.4.2C The TOE shall be labelled with its reference.

ACM_CAP.4.3C The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.

ACM_CAP.4.4C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.4.5C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.4.6C The CM system shall uniquely identify all configuration items.

ACM_CAP.4.7C The CM plan shall describe how the CM system is used.

ACM_CAP.4.8C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.4.9C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.4.10C The CM system shall provide measures such that only authorised changes are made to the configuration items.

ACM_CAP.4.11C The CM system shall support the generation of the TOE.

ACM_CAP.4.12C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

5.4.1.3 ACM_SCP.2 – Problem tracking CM coverage

ACM_SCP.2.1D The developer shall provide CM documentation.

ACM_SCP.2.1C The CM documentation shall show that the CM system, as a minimum, tracks the following: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws.

ACM_SCP.2.2C The CM documentation shall describe how configuration items are tracked by the CM system.

5.4.2 Delivery and Operation (ADO)

5.4.2.1 ADO_DEL.2 – Detection of modification

ADO_DEL.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.2.2D The developer shall use the delivery procedures.

ADO_DEL.2.1C 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.2.2C The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO_DEL.2.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

5.4.2.2 ADO_IGS.1 – Installation, generation, and start-up procedures

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 documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.

5.4.3 Development (ADV)

5.4.3.1 ADV_FSP.2 – Fully defined external interfaces

ADV_FSP.2.1D The developer shall provide a functional specification.

ADV_FSP.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.2.2C The functional specification shall be internally consistent.

ADV_FSP.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

ADV_FSP.2.4C The functional specification shall completely represent the TSF.

ADV_FSP.2.5C The functional specification shall include rationale that the TSF is completely represented.

5.4.3.2 ADV_HLD.2 – Security enforcing high-level design

ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

ADV_HLD.2.1C The presentation of the high-level design shall be informal.

ADV_HLD.2.2C The high-level design shall be internally consistent.

ADV_HLD.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.5C 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.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_HLD.2.9C The high-level design shall describe the separation of the TOE into TSPenforcing and other subsystems.

5.4.3.3 ADV_IMP.1 – Subset of the implementation of the TSF

ADV_IMP.1.1D The developer shall provide the implementation representation for a selected subset of the TSF.

ADV_IMP.1.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2C The implementation representation shall be internally consistent.

5.4.3.4 ADV_LLD.1 – Descriptive low-level design

ADV_LLD.1.1D The developer shall provide the low-level design of the TSF.

ADV_LLD.1.1C The presentation of the low-level design shall be informal.

ADV_LLD.1.2C The low-level design shall be internally consistent.

ADV_LLD.1.3C The low-level design shall describe the TSF in terms of modules.

ADV_LLD.1.4C The low-level design shall describe the purpose of each module.

ADV_LLD.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV_LLD.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_LLD.1.10C The low-level design shall describe the separation of the TOE into TSP enforcing and other modules.

5.4.3.5 ADV_RCR.1 – Informal correspondence demonstration

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.

5.4.3.6 ADV_SPM.1 – Informal TOE security policy model

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.

ADV_SPM.1.1C The TSP model shall be informal.

ADV_SPM.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

ADV_SPM.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

5.4.4 Guidance Documents (AGD)

5.4.4.1 AGD_ADM.1– Administrator guidance

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.

5.4.4.2 AGD_USR.1 – User guidance

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.

5.4.5 Life Cycle Support (ALC)

5.4.5.1 ALC_DVS.1 – Identification of security measures

ALC_DVS.1.1D The developer shall produce development security documentation.

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

5.4.5.2 ALC_LCD.1 – Developer defined life-cycle model

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

5.4.5.3 ALC_TAT.1 – Well-defined development tools

ALC_TAT.1.1D The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of the development tools.

ALC_TAT.1.1C All development tools used for implementation shall be well defined.

ALC_TAT.1.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

5.4.6 Test (ATE)

5.4.6.1 ATE_COV.2 – Analysis of coverage

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

5.4.6.2 ATE_DPT.1 – Testing: high-level design

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

5.4.6.3 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, 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.

5.4.6.4 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.

5.4.7 Vulnerability (AVA)

5.4.7.1 AVA_MSU.2 – Validation of analysis

AVA_MSU.2.1D The developer shall provide guidance documentation.

AVA_MSU.2.2D The developer shall document an analysis of the guidance documentation.

AVA_MSU.2.1C The guidance documentation 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.

AVA_MSU.2.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.2.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.2.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.2.5C The analysis documentation shall demonstrate that the guidance documentation is complete.

5.4.7.2 AVA_SOF.1 – Strength of TOE security function evaluation

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.

5.4.7.3 AVA_VLA.2 – Independent vulnerability analysis

AVA_VLA.2.1D The developer shall perform and document an analysis of the TOE deliverables searching for ways in which a user can violate the TSP.

AVA_VLA.2.2D The developer shall document the disposition of identified vulnerabilities.

AVA_VLA.2.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.2.2C The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

5.5 Security Requirements for the IT Environment

There are no security requirements for the IT environment.

6 TOE Summary Specification

6.1 TOE IT Security Functions

This section presents a high-level summary of the IT security functions performed by the TOE and provides a mapping between the identified security functions and the Security Functional Requirements that it must satisfy.

IT Security Function	Security Functional Requirements	Description
F.AUDIT	FAU_GEN.1.1 FAU_GEN.1.2 FAU_SAR.1.1 FAU_SAR.1.2 FPT_STM.1.1	Audit data is generated only within the CypherNET encryptor, and stored in an audit table in non-volatile memory. All auditable events are associated with operations that occur in CypherNET only, thus there is no requirement for audit logs on CypherManager. The TOE is able to generate an audit record for each of the auditable events listed in FAU_GEN.1.1 and FAU_GEN.1.2. The CypherNet encryptor has a Real Time Clock (RTC) from which a timestamp is obtained to record within each audit record. Authorised users can view the audit log, using SNMPv3 remote management from CypherManager or through the console port. In each case, the user is identified and authenticated before access is granted to the audit log. In each case, the data is presented in a human readable format, with CypherManager and the console mode presenting the data as a scrolled list of audit text. The audit log has a finite size for logging audit records. Once this space has been used, the audit log is either cycled back around, or disabled as selected by the Administrator. Alternatively, the Administrator is permitted to clear the audit log at any time.

IT Security Function	Security Functional Requirements	Description
F.CERTIFICATE_ MANAGEMENT	FCS_COP.1.1.C FCS_COP.1.1.F FCS_COP.1.1.G FDP_DAU.1.1 FDP_DAU.1.2 FTP_ITC.1.1 FTP_ITC.1.2 FTP_ITC.1.3	The TOE shall manage all necessary tasks to support X.509 certificate based authentication. These tasks are: a. Generating and installing signed X.509 certificates into CypherNET b. Authenticating received X.509 certificates using installed trusted CA root certificates Operations relating to generating, X.509 certificates require the use of the RSA algorithm to generate the private and public key pair (FCS_COP.1.1.C). X.509 certificate signing operations are done using the RSA (FCS_COP.1.1.G) signature algorithm. When CypherManager requests a new public key from an encryptor, CypherNET hashes the data that will be returned using SHA-1 (FCS_COP.1.1.F) to create a validation code (FDP_DAU.1.1). The validation code is displayed on the front panel of the encryptor. (FDP_DAU.1.2). CypherManager also hashes the received data and displays the validation code. Both the CypherManager user and the remote operator must agree that the validation codes are the same before the CypherManager user signs the X.509 certificate. When CypherManager returns the signed certificate the same process is repeated again with the CypherManager user and remote operator agreeing that the validation codes are the same before the X.509 certificate is loaded into the encryptor. The Encryptor uses the certificate to establish a trusted communications channel between itself and other Encryptors (remote trusted IT products). Both encryptors must have a valid X.509 certificate, which has been signed by a trusted CA, to protect the confidentiality and integrity of transmitted information and is logically distinct from other channels.

IT Security Function	Security Functional Requirements	Description
F.DATA_EXCHANGE	FCS_COP.1.1.A FCS_COP.1.1.B FDP_UCT.1.1	The TOE encrypts the payload or the bit stream on the basis of the address in the cell, frame, line/path number or channel number and whether the CAT entry requires encryption of traffic on that address or channel number. If encryption is required, CypherNET performs hardware-or software based 168 bit DES encryption or 128 or 256 bit AES encryption in CFB or counter mode on the cell or Ethernet frame payload or CFB mode on the bit stream or 256 bit AES encryption on the SONET/SDH payload.
F.IDENTIFICATION	FIA_AFL.1.1 FIA_AFL.1.2 FIA_UAU.2.1 FIA_UID.2.1	To modify and view any of the security attributes of the TOE, authorised users must identify and authenticate via one of two mechanisms depending on whether they are using the SNMPv3 functionality or the console management functionality. Identification & Authentication services are only performed by the CypherNET encryptor. All user passwords must have a minimum length of eight characters. The set of possible characters are A-Z, a-z, 0-9 and ` ~ ! @ # $ % ^ & * ( ) _ - + = { [}] : ; ” ’ , < . > ? / | \\. For local management using the local console port of the CypherNET encryptor, users logon to the CypherNET by supplying a user ID and their authentication password. The CypherNET then compares the user ID and the password supplied with the local authentication password. If the authentication password does not match, for that user ID in the CypherNET User Account Table, then identification and authentication fails, the console session is not started, and the event is audited. After three consecutive unsuccessful logon attempts the user account will be disabled for three minutes. If the user ID and authentication password match the entry in the user table, a console session is opened. For remote management using SNMPv3 the CypherManager remote management station will generate an appropriate authentication key, used to authenticate the

IT Security Function	Security Functional Requirements	Description
		remote management data, and a privacy key used to 56-bit DES encrypt the remote management data. Both keys are generated on CypherManager after retrieving the SNMPv3 Engine ID of the CypherNET encryptor and the user supplied passwords. The remote management data is associated with a user ID entered by the user on CypherManager to make the SNMPv3 packet. The authenticated (and optionally encrypted) SNMPv3 packets are then sent to CypherNET. The User ID, local privacy passwords and local authentication passwords are stored within the User Account Table of CypherNET, with the first administrator account being created during the initialisation of the encryptor. If CypherNET cannot decrypt the data or the authentication process as specified in RFC2574 fails, then the identification and authentication of that SNMPv3 data fails, the SNMPv3 data is discarded, and the event is audited. Each SNMPv3 packet received is identified and authenticated in this way.

IT Security Function	Security Functional Requirements	Description
F.KEY_ MANAGEMENT	FCS_CKM.1.1.A FCS_CKM.1.1.B FCS_CKM.1.1.C FCS_CKM.2.1.A FCS_CKM.4.1 FCS_COP.1.1.C FMT_MSA.2.1.C	The TOE shall manage all the necessary keys and mechanisms to support its cryptographic operations, namely: a. Generating RSA public/private key pairs for both CypherManager and CypherNET. b. Generating and securely transferring master keys between encryptors. Keys are distributed between encryptors using RSA public key cryptography and X.509 certificates are used for authentication; c. Updating session keys used for AES/DES encryption between encryptors. AES/DES session keys are periodically updated according to local security policy requirements set by Administrators or Supervisors. d. Generating privacy DES keys and authentication keys for SNMPv3 management. e. Protecting user passwords used for generating privacy DES keys and authentication keys, during user account setup on a CypherNET, by encrypting the password data with the master CSP key of the intended CypherNET encryptor that will operate the user account. f. Ensuring all generated keys are random. g. Session keys held in volatile memory (RAM) are erased on loss of power.

IT Security Function	Security Functional Requirements	Description
F.INFORMATION_ FLOW_ CONTROL	FDP_IFC.1.1 FDP_IFF.1.1 FDP_IFF.1.2 FDP_IFF.1.3 FDP_IFF.1.4 FDP_IFF.1.5 FDP_IFF.1.6 FMT_MSA.2.1.A FMT_MSA.3.1.A FMT_MSA.3.2.A	The TOE shall control the flow of SONET/SDH frames, cells, Ethernet frames or bit streams received on the private network interface and on the public network interface from external hosts on the basis of the address in the cell, Ethernet frame, SONET/SDH line/path number or bit stream channel number. In doing so, the TOE shall take one of four possible actions, encrypt/decrypt the payload or bit stream, using DES or AES in CFB or counter mode, pass the payload or bit stream unchanged, discard the payload or bit stream. The TOE determines the appropriate action to take on any given cell, frame or bit stream by examining the list of entries in the CAT. By default, for a given address, line/path number or channel number that is not listed in the CAT, the cell, frame or bit stream is discarded.
F.ROLE_ BASED_ ACCESS	FDP_ACC.1.1 FDP_ACF.1.1 FDP_ACF.1.2 FDP_ACF.1.3 FDP_ACF.1.4 FMT_MSA.1.1.B FMT_MSA.2.1.B FMT_MSA.3.1.B FMT_MSA.3.2.B FMT_MTD.1.1 FMT_SMR.1.1 FMT_SMR.1.2 FTA_SSL.3.1 FMT_MSA.1.1.A	The TOE can be accessed using SNMPv3 packets received on the Ethernet management port interface and the local and network interfaces or via the console management port interface. The encryptor's USB port can be used to backup and restore configuration data and to upgrade firmware. Users will be allowed access to the TOE when a valid user id, password are provided. Additionally, any packets or sessions (i.e. SNMPv3) must be properly authenticated and for access to be obtained. SNMPv3 uses a privacy password that is associated with the user id to optionally encrypt/decrypt the packets. If any of these conditions are not met then access will be denied. The TOE defines three roles for accessing the TSFs. These are: Administrators: Who can change defaults, query, modify, delete and clear the CAT and the CAT information flow address and actions, User accounts, activate X.509 certificates, clear the audit log, view the audit log, set the system time and backup and restore the encryptor configuration data and

IT Security Function	Security Functional Requirements	Description
		remotely upgrade the firmware. Supervisors: Who can change defaults, query, modify, delete and clear the CAT, view the User accounts table and audit log and set the system time. Operators: Who can query the CAT and User Account tables only, and view the audit log. When the TOE is accessed the TOE associates users with these roles and prevents a user from performing operations on the TSF's that they are not authorised to perform. The user can only enter values for attributes that will keep the TOE in a secure state. The console user session will be automatically terminated by CypherNET after a period of 10 minutes as a result of user inactivity.
F.SECURE_ REMOTE_ MANAGEMENT	FPT_ITT.1.1 FCS_COP.1.1.A	The TOE shall protect the confidentiality of remote management data between the CypherNET encryptors and the CypherManager remote management station. The TOE can encrypt SNMPv3 data packets using 56-bit DES keys that are derived from the Engine ID of the CypherNET being managed and the user's privacy password. The user initiates the remote management session by executing the CypherManager software on their workstation.

IT Security Function	Security Functional Requirements	Description
F.SELF_ PROTECT	FCS_CKM.4.1 FPT_AMT.1.1 FPT_FLS.1.1 FPT_PHP.3.1.A FPT_PHP.3.1.B FPT_TST.1.1 FPT_TST.1.2 FPT_TST.1.3 FCS_COP.1.1.A	The TOE protects itself from attempts to get access to the user passwords and key material stored within CypherNET. An erase mechanism is provided that is activated whenever the case is opened. Once activated, the master key is erased from battery-backed volatile memory. The master key encrypts all private key material and user password data, and so removal of the master key means the encrypted data cannot be accessed. CypherNET performs self-tests during start-up to check that the underlying functionality of the TSF is functioning correctly. The tests include verification of the cryptographic processors, Random Noise Source, Firmware integrity, System Memory, Software integrity, as well as TSF configuration data. The results of the self-tests are audited. If any of the self-tests fail then the TOE will preserve a secure state and all output is suppressed. The TOE protects its own private key on CypherManager by encrypting the private key using triple DES and a passphrase. Only a user who has access to the passphrase can unlock the private key of the CypherManager.

Table 11 – TOE IT Security Functions

The set of high level IT security functions map directly onto functions derived from the Common Criteria Part 2 Security Functional Requirements. These CC Part 2 requirements have been used in the subsequent analyses and mapping’s to demonstrate suitability and mutual support of TOE security functions. Therefore, it is possible to trace how these high level functions contribute to satisfying the TOE SFRs.

6.2 TOE Assurance Measures

The TOE satisfies the CC EAL4 assurance requirements. Senetas has assurance measures for the TOE to satisfy the stated SARs. The following table shows which assurance measures are traced to the assurance requirements identified in Section 5.3.

No.	Component	Assurance Measure
Class ACM: Configuration management		
1	ACM_AUT.1	Practices and procedures for configuration management of all TOE components. An automated tool (Aegis) is used to partially automate the CM process.
2	ACM_CAP.4	Practices and procedures for the TOE generation and acceptance procedures. A Configuration Management plan, Configuration List and Acceptance Plan are used to satisfy the requirements of this assurance measure.
3	ACM_SCP.2	Practices and procedures for problem tracking. An automated tool is used to track problems in the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation and security flaws.
Class ADO: Delivery and operation		
4	ADO_DEL.2	Delivery procedures for the secure delivery of the TOE to the end user
5	ADO_IGS.1	Detailed installation, generation and start-up procedures
Class ADV: Development		
6	ADV_FSP.2	Functional Specification document
7	ADV_HLD.2	High-Level Design document
8	ADV_IMP.1	A representative sample of the circuit diagrams and source code
9	ADV_LLD.1	Low-Level Design document
10	ADV_RCR.1	Representational Correspondence document
11	ADV_SPM.1	Informal TOE security policy model document
Class AGD: Guidance documents		
12	AGD_ADM.1	Administrator Guidance manual
13	AGD_USR.1	User Guidance manual
Class ALC: Life cycle support		
14	ALC_DVS.1	Company Security Policy document
15	ALC_LCD.1	Life-cycle model document

16	ALC_TAT.1	Development tool documentation. Only well defined languages are used for the implementation – C/C++ and ASN.1
Class ATE: Tests		
17	ATE_COV.2	Test Plan document and coverage analysis
18	ATE_DPT.1	Acceptance and developer testing and a depth analysis
19	ATE_FUN.1	Acceptance and developer testing
20	ATE_IND.2	Evaluator responsibility
Class AVA: Vulnerability assessment		
21	AVA_MSU.2	Validation of analysis document
22	AVA_SOF.1	Strength of TOE security function document
23	AVA_VLA.2	Vulnerability analysis document

Table 12 – TOE Assurance Measures

7 Rationale

7.1 Security Objectives Rationale

7.1.1 Mapping of Threats, OSPs and Assumptions to Security Objectives

The following table demonstrates that the each threat, OSP and assumption is addressed by at least one security objective, and each security objective addresses at least one threat, OSP or assumption.

Objectives **Assumptions, Threats, OSPs**	**O.ADMIN**	**O.AUDIT**	**O.AUDITLOG**	**O.AUTHDATA**	**O.CERTGEN**	**O.CONNECT**	**O.ENCRYPT**	**O.FAILSAFE**	**O.INFOFLOW**	**O.IDENT**	**O.INSTALL**	**O.KEYMAN**	**O.PERSONNEL**	**O.PHYSICAL**	**O.REMOTEMGT**	**O.ROLES**	**O.TAMPER**	**O.ROLEMGT**
ASSUMPTIONS																		
A.ADMIN													✓					
A.AUDIT			✓															
A.CYPHERMANAGER											✓			✓				
A.INSTALL											✓							
A.LOCATE											✓			✓				
A.PRIVATEKEY				✓														
THREATS																		
T.ABUSE		✓	✓	✓						✓			✓			✓		✓
T.ATTACK	✓	✓	✓					✓								✓		✓

Objectives **Assumptions, Threats, OSPs**	**O.ADMIN**	**O.AUDIT**	**O.AUDITLOG**	**O.AUTHDATA**	**O.CERTGEN**	**O.CONNECT**	**O.ENCRYPT**	**O.FAILSAFE**	**O.INFOFLOW**	**O.IDENT**	**O.INSTALL**	**O.KEYMAN**	**O.PERSONNEL**	**O.PHYSICAL**	**O.REMOTEMGT**	**O.ROLES**	**O.TAMPER**	**O.ROLEMGT**
T.CAPTURE							✓		✓									
T.CONNECT					✓	✓			✓			✓						
T.IMPERSON	✓	✓	✓	✓						✓								
T.LINK					✓		✓		✓			✓						
T.MAL								✓										
T.OBSERVE												✓			✓			
T.PHYSICAL											✓		✓	✓			✓	
T.PRIVILEGE		✓	✓	✓						✓			✓			✓		✓
OSP'S																		
P.CRYPTO					✓		✓					✓			✓			
P.INFOFLOW	✓								✓									
P.ROLES	✓															✓		✓

Table 13 – Mapping of Threats, OSPs and Assumptions to Security Objectives

7.1.2 Informal argument of adequacy and correctness of mapping

7.1.2.1 Assumptions

Assumption	Description
A.ADMIN	*O.PERSONNEL* ensures that only trusted and competent administrators are authorised to manage the TOE.
A.AUDIT	*O.AUDITLOG* ensures that the facilities to effectively manage audit information are provided.
A.CYPHERMANAGER	*O.INSTALL* ensures that the CypherManager Management Station is installed and managed in a secure environment. *O.PHYSICAL* ensures that the CypherManager Management Station will be protected from physical attacks The combination of these objectives will prevent unauthorised users from attempting to compromise the security functions of the CypherManager Management Station and therefore cover this assumption.
A.INSTALL	*O.INSTALL* ensures that the TOE is delivered, installed, managed and operated in a manner that maintains security.
A.LOCATE	*O.INSTALL* ensures that CypherNET is installed correctly in a secure environment while *O.PHYSICAL* ensures that this environment remains secure from unauthorised people.
A.PRIVATEKEY	*O.AUTHDATA* ensures that the authentication data for each account on the TOE is held securely and not disclosed to persons unauthorised to use that account. The authentication data includes the password to protect the CypherManager's private key.

Table 14 – Informal argument of assumptions

7.1.2.2 Threats

Threat	Justification

Threat	Justification
T.ABUSE	*O.AUDIT* provides a means of recording security relevant events and *O.AUDITLOG* ensures that the facilities to effectively manage audit information are provided. This allows authorised users to detect modifications. This will prevent compromises being undetected. *O.ROLES* ensures the user can only access the operations that the role authorises. *O.ROLEMGT* ensures that users are allocated roles with *least privilege*. This can minimise the threat damage caused by the role. *O.IDENT* ensures that all users are uniquely identified and authenticated before access to TOE management features is allowed. *O.AUTHDATA* ensures that the authentication data for each account on the TOE is held securely and not disclosed to persons unauthorised to use that account. So if the audit trail indicates an abuse by a certain role, then the human allocated that role can be held responsible for those actions. This in conjunction with abuse detection (*O.AUDIT* and *O.AUDITLOG*) will deter users from intentionally abusing their privileges. *O.PERSONNEL* supports the above objectives by ensuring that only trusted and competent personnel operate the TOE. A trusted user will not intentionally abuse their privileges, while a competent user will not accidentally perform operations compromising information. The combination of these objectives will reduce this threat to an acceptable level.

Threat	Justification
T.ATTACK	*O.AUDIT* provides a means of recording security relevant events and *O.AUDITLOG* ensures that the facilities to effectively manage audit information are provided. This allows authorised users to detect modifications. This will prevent compromises being undetected. *O.ROLES* ensures the user can only access the operations that the role authorises. *O.ROLEMGT* ensures that users are allocated roles with *least privilege*. This prevents insider users from doing operations for which they are not authorised. *O.ADMIN* ensures that only authorised users can access the TOE management functions. This prevents outsider attackers from accessing the TOE management functions and compromising information. *O.FAILSAFE* ensures that if an error occurs the TOE will preserve a secure state. If a logical attack results in an error condition, then the TOE will not compromise information. The combination of these objectives is sufficient to reduce undetected logical attacks from insiders and outsiders to an acceptable level.
T.CAPTURE	*O.INFOFLOW* allows for selected SONET/SDH frames, ATM cells, or Ethernet frames to be encrypted or discarded according to a defined security policy and therefore preventing capture on the public network. *O.ENCRYPT* allows for the encryption of SONET/SDH, ATM or Ethernet payloads or bit streams ensuring that captured data can not be readable without private keys. When these objectives are met, the threat of confidential information being recovered by an attacker will suitably diminish.

Threat	Justification
T.CONNECT	*O.INFOFLOW* allows authorised users to explicitly allow connections, however, by default all connections, other than SONET/SDH management channels, ATM management cells or Ethernet management frames to the TOE, will be discarded. *O.KEYMAN* ensures that encrypted connections cannot be made unless the originator and receiver hold a valid X.509 certificate signed by a trusted CA. This will prevent connections with untrusted networks from being established. *O.CERTGEN* supports *O.KEYMAN* by ensuring the TOE has the capability to generate, issue and manage X.509 certificates. *O.CONNECT* supports the environment to ensure that connections that would undermine security are not established by those responsible for the TOE. When all these objectives are met, the threat of an insecure connection being created by an attacker will be suitably diminished.
T.IMPERSON	*O.IDENT* uniquely identifies all users and authenticates the claimed identity before granting a user access to the TOE management facilities. For an attacker to impersonate an authorised user, the attacker must know the user's identity and authentication data. To restrict opportunities for impersonation attacks accounts are disabled on authentication failure *O.AUTHDATA* ensures that users are responsible not to disclose their authentication data so attackers cannot impersonate authorised users. *O.ADMIN* ensures only authorised users can manage the TOE and its security features. *O.AUDIT* provides a means of recording security relevant events and *O.AUDITLOG* ensures that the facilities to effectively manage audit information are provided. This allows authorised users to detect when impersonation attacks (eg. brute force password guessing) occur. When all these objectives are met, the threat of privileged users being impersonated by an inside or outside attacker will suitably diminish.

Threat	Justification
T.LINK	*O.INFOFLOW* allows authorised users to explicitly allow connections, however, by default, all connections to the TOE will be discarded. *O.ENCRYPT* allows for the encryption of SONET/SDH, ATM, Ethernet payloads or bit streams. *O.KEYMAN* provides the means for exchanging keys with only other authorised encryptors to establish a link. The other encryptors are only authorised due to X.509 certificate attributes as provided by *O.CERTGEN*. So *O.KEYMAN* and *O.CERTGEN* restrict the number of possible communications paths to only other authorised encryptors. The objectives *O.INFOFLOW, O.KEYMAN* and *O.CERTGEN* combine to minimise the number of communication links that an encryptor will have. The minimal links will reduce the opportunity an attacker has to deduce information. As confidential information over these links will be encrypted due to *O.ENCRYPT*, the attacker will require more resources and knowledge to deduce any useful information. Therefore the combination of all these objectives will lower this threat to an acceptable level.
T.MAL	*O.FAILSAFE* ensures that the TOE will enter a secure state if any malfunction of the TOE is detected.
T.OBSERVE	*O.REMOTEMGT* ensures that remote management sessions can be encrypted. This will minimise the threat that an attacker may observe legitimate management communications, as the data would have to be decrypted with secret session keys. *O.KEYMAN* supports *O.REMOTEMGT* to allow cryptographic key management to enable cryptographic exchanges between the CypherNet encryptor and CypherManager. When all these objectives are met, the threat of legitimate management communications being observed by an attacker will be suitably diminished.

Threat	Justification
T.PHYSICAL	*O.INSTALL* ensures that the TOE is delivered, installed, managed, and operated in a manner, which maintains IT security. *O.PHYSICAL* ensures that those parts of the TOE that are critical to security policy enforcement are protected from physical attack. *O.PERSONNEL* ensures that those responsible for the TOE are competent to manage the TOE and can be trusted not to deliberately abuse their privileges. The above environmental objectives provide a secure environment for the TOE to reduce a physical attack from occurring. *O.TAMPER* provides physical protection of stored assets (user authentication and cryptography key material) to prevent a security compromise via physical means if the above environmental measures are not sufficient. With all objectives met, this threat is removed.
T.PRIVILEGE	*O.ROLES* ensures the user can only access the operations that the role authorises. *O.ROLEMGT* ensures that users are allocated roles with *least privilege*. This limits the operations and therefore the damage a compromise can lead to. *O.PERSONNEL* ensures that users within the environment are trusted and competent. This will minimise the threats from hostile or wilfully negligent administrators. *O.IDENT* ensures that a user requesting information is correctly identified. While *O.AUTHDATA* ensures that they are responsible with that information by not disclosing it to users so those people authorised to use the account can be held responsible for their actions. *O.AUDIT* provides a means of recording security relevant events and *O.AUDITLOG* ensures that the facilities to effectively manage audit information are provided. This allows authorised users to monitor possible changes to the configuration of the TOE, allowing all authorised users to detect modifications. The user’s identity from *O.IDENT* will be recorded in the audit log, so privileged users will have their actions recorded and reviewed to deter them from abusing their privileges. When all these objectives are met, the threat of privileged users compromising information is suitably diminished.

Table 15 – Informal argument of threats

7.1.2.3 Policies

Policy	Description
P.CRYPTO	*O.ENCRYPT*, *O.KEYMAN, O.REMOTEMGT* and *O.CERTGEN* provide the confidentiality, authentication and key management services specified by this organisational security policy.
P.INFOFLOW	*O.INFOFLOW* provides the traffic flow control specified in the organisational security policy. *O.ADMIN* ensures that only authorised users can set the traffic control as specified in the organisational security policy.
P.ROLES	*O.ROLEMGT* ensures that administrators will allocate users to distinct roles on the basis of least privilege. *O.ROLES* ensures that users can only perform the operations for which their role is explicitly authorised. *O.ADMIN* ensures that only authorised users can manage the TOE as specified in the organisational security policy.

Table 16 – Informal argument of policies

7.1.2.4 Rationale

Given the arguments in the above tables and the mapping's shown in Table 13, it has been demonstrated that the security objectives are suitable to counter all threats and to consider all assumptions and organisational security policies.

7.2 Security Requirements Rationale

7.2.1 Mapping of Security Functional Requirements to Security Objectives

The following table demonstrates that the each TOE SFR is mapped to at least one TOE security objective.

Security Objective Security Functional Requirement	O.ADMIN	O.AUDIT	O.CERTGEN	O.ENCRYPT	O.FAILSAFE	O.INFOFLOW	O.IDENT	O.KEYMAN	O.REMOTEMGT	O.ROLES	O.TAMPER
FAU_GEN.1.1		✓									
FAU_GEN.1.2		✓									
FAU_SAR.1.1		✓									
FAU_SAR.1.2		✓									
FCS_CKM.1.1.A								✓			
FCS_CKM.1.1.B								✓			
FCS_CKM.1.1.C								✓			
FCS_CKM.2.1.A								✓			
FCS_CKM.4.1								✓			✓
FCS_COP.1.1.A				✓					✓		✓
FCS_COP.1.1.B				✓							
FCS_COP.1.1.C			✓					✓			
FCS_COP.1.1.F			✓								
FCS_COP.1.1.G			✓								
FDP_ACC.1.1	✓										
FDP_ACF.1.1	✓										
FDP_ACF.1.2	✓										
FDP_ACF.1.3	✓										
FDP_ACF.1.4	✓										
FDP_DAU.1.1			✓								
FDP_DAU.1.2			✓								
FDP_IFC.1.1						✓					
FDP_IFF.1.1						✓					
FDP_IFF.1.2						✓					
FDP_IFF.1.3						✓					
FDP_IFF.1.4						✓					
FDP_IFF.1.5						✓					

Security Objective **Security Functional Requirement**	**O.ADMIN**	**O.AUDIT**	**O.CERTGEN**	**O.ENCRYPT**	**O.FAILSAFE**	**O.INFOFLOW**	**O.IDENT**	**O.KEYMAN**	**O.REMOTEMGT**	**O.ROLES**	**O.TAMPER**
FDP_IFF.1.6						✓					
FDP_UCT.1.1				✓							
FIA_AFL.1.1							✓				
FIA_AFL.1.2							✓				
FIA_UAU.2.1							✓				
FIA_UID.2.1							✓				
FMT_MSA.1.1.A						✓					
FMT_MSA.1.1.B										✓	
FMT_MSA.2.1.A						✓					
FMT_MSA.2.1.B										✓	
FMT_MSA.2.1.C								✓			
FMT_MSA.3.1.A						✓					
FMT_MSA.3.1.B										✓	
FMT_MSA.3.2.A						✓					
FMT_MSA.3.2.B										✓	
FMT_MTD.1.1	✓									✓	
FMT_SMR.1.1										✓	
FMT_SMR.1.2										✓	
FPT_AMT.1.1					✓			✓			
FPT_FLS.1.1					✓						
FPT_ITT.1.1									✓		
FPT_PHP.3.1.A											✓
FPT_PHP.3.1.B											✓
FPT_STM.1.1		✓									
FPT_TST.1.1					✓			✓			
FPT_TST.1.2					✓			✓			
FPT_TST.1.3					✓			✓			
FTA_SSL.3.1	✓										
FTP_ITC.1.1			✓								
FTP_ITC.1.2			✓								
FTP_ITC.1.2			✓								

Table 17 – Mapping of Security Functional Requirements to Security Objectives

7.2.2 Informal Argument of Sufficiency

The following table contains a justification for the chosen SFRs and their suitability to satisfy each security objective for the TOE.

Objective	Security Functional Requirement	Justification
O.ADMIN	FDP_ACC.1.1 FDP_ACF.1.1 FDP_ACF.1.2 FDP_ACF.1.3 FDP_ACF.1.4 FTA_SSL.3.1 FMT_MTD.1.1	*FDP_ACC.1.1*, *FDP_ACF.1.1*, *FDP_ACF.1.2*, *FDP_ACF.1.3* and *FDP_ACF.1.4* together provide the capability for management of the TOE security functions by authorised users in a manner required for correct operation and management of the TOE as required by *O.ADMIN*. *FTA_SSL.3.1* provide additional protection automatically terminating management sessions after a period of user inactivity. *FMT_MTD.1.1* provides the function so authorised roles can manage the TSF data.
O.AUDIT	FAU_GEN.1.1 FAU_GEN.1.2 FAU_SAR.1.1 FAU_SAR.1.2 FPT_STM.1.1	*FAU_GEN.1.1* and *FAU_GEN.1.2* provide the capability for generating and recording audit events in the manner required by *O.AUDIT*. *FAU_SAR.1.1* and *FAU_SAR.1.2* provide the capability for viewing audit logs to support the effective use and management of the audit facilities in a manner required by *O.AUDIT*. *FPT_STM.1.1* ensures that a date and time stamp is recorded with the audit record.

Objective	Security Functional Requirement	Justification
O.CERTGEN	FCS_COP.1.1.C FCS_COP.1.1.F FCS_COP.1.1.G FDP_DAU.1.1 FDP_DAU.1.2 FTP_ITC.1.1 FTP_ITC.1.2 FTP_ITC.1.3	*FCS_COP.1.1.C* uses the RSA algorithm to encrypt the RSA private key for X.509 certificates. *FCS_COP.1.1.G* together with *FCS_COP.1.1.F* provides the means for signing completed X.509 certificates for CypherNET. These cryptographic functions meet the standards required by FIPS 140-2 and ACSI33. *FDP_DAU.1.1* and *FDP_DAU.1.2* provides the means for producing a digest of the data for authentication purposes, when generating partial X.509 certificates in certificate load mode, and after sending completed and signed X.509 certificates from CypherManager to the CypherNET encryptor. *FTP_ITC.1.1*, *FTP_ITC.1.2* and *FTP_ITC.1.3* provides the means for using the X.509 certificates to authenticate other Encryptors and establish a secure trusted channel.
O.ENCRYPT	FCS_COP.1.1.A FCS_COP.1.1.B FDP_UCT.1.1	*FCS_COP.1.1.A, FCS_COP.1.1.B and , FDP_UCT.1.1*, together provide the capability for encrypting information to protect the confidentiality of information transferred across the SONET/SDH, ATM, Ethernet or data networks, as required by O.ENCRYPT. The cryptographic functions meet the standards required by FIPS 140-2 and ACSI33.
O.FAILSAFE	FPT_AMT.1.1 FPT_FLS.1.1 FPT_TST.1.1 FPT_TST.1.2 FPT_TST.1.3	*FPT_FLS.1.1* together with *FPT_AMT.1.1, FPT_TST.1.1, FPT_TST.1.2* and *FPT_TST.1.3* provides the capability for the TOE to demonstrate correct operation by performing self-tests on start-up which ensures that the TOE will enter a secure state if any internal failure is detected.

Objective	Security Functional Requirement	Justification
O.INFOFLOW	FDP_IFC.1.1 FDP_IFF.1.1 FDP_IFF.1.2 FDP_IFF.1.3 FDP_IFF.1.4 FDP_IFF.1.5 FDP_IFF.1.6 FMT_MSA.1.1.A FMT_MSA.2.1.A FMT_MSA.3.1.A FMT_MSA.3.2.A	*FDP_IFC.1.1, FDP_IFF.1.1, FDP_IFF.1.2, FDP_IFF.1.3, FDP_IFF.1.4, FDP_IFF.1.5, FDP_IFF.1.6, FMT_MSA.1.1.A, FMT_MSA.2.1.A, FMT_MSA.3.1.A* and *FMT_MSA.3.2.A* together provide the capability for authorised users to control traffic flow between subjects using the ATM VPI/VCI address, the Ethernet MAC address, the line/path number for a SONET/SDH connection and the channel number for a bit stream connection in a manner required by O.INFOFLOW.
O.IDENT	FIA_UAU.2.1 FIA_UID.2.1 FIA_AFL.1.1 FIA_AFL.1.2	*FIA_UAU*.2.1 and *FIA_UID.2.1* provide the capability for identifying and authenticating all users in a manner required by O.IDENT. *FIA_AFL.1.1* and *FIA_AFL.1.2* provide additional protection by limiting the number of unsuccessful authentication attempts before imposing a timeout on that user account.
O.KEYMAN	FCS_COP.1.1.C FCS_CKM.1.1.A FCS_CKM.1.1.B FCS_CKM.1.1.C FCS_CKM.2.1.A FCS_CKM.4.1 FMT_MSA.2.1.C	*FCS_CKM.1.1.A, FCS_CKM.1.1.B, FCS_CKM.1.1.C, FCS_CKM.2.1.A,* and *FCS_CKM.4.1* provide the capability for generating, distributing and destroying cryptographic keys as required to provide means for exchanging keys with an authorised TOE as required by O.KEYMAN. FCS_COP.1.1.C provides RSA encryption of session keys. *FMT_MSA.2.1.C* ensures that only secure cryptographic key values are generated. These cryptographic functions meet the standards required by FIPS 140-2 and ACSI33.

Objective	Security Functional Requirement	Justification
O.REMOTEMGT	FCS_COP.1.1.A FPT_ITT.1.1	*FCS_COP.1.1.A,* provides the capability for encryption methods for management data over the network. *FPT_ITT.1.1* ensures the confidentiality of remote management information is maintained.
O.ROLES	FMT_MSA.1.1.B FMT_MSA.2.1.B FMT_MSA.3.1.B FMT_MSA.3.2.B FMT_MTD.1.1 FMT_SMR.1.1 FMT_SMR.1.2	*FMT_SMR.1.1* specifies the three possible roles administrator, supervisor and operator. *FMT_MSA.1.1.B, FMT_MSA.2.1B FMT_MSA.3.1.B, FMT_MSA.3.2.B* defines each role's privileges for managing the TSF security attributes. *FMT_MTD.1.1* defines each role's privileges for managing the TSF data. *FMT_SMR.1.2* associates a human with one role. In combination, these SFRs restricts the human's access to only those TSF attributes, data and operations explicitly allowed by the associated role.
O.TAMPER	FPT_PHP.3.1.A FPT_PHP.3.1.B FCS_COP.1.1.A FCS_CKM.4.1	*FPT_PHP.3.1.A and FPT_PHP.3.1.B* provides the capability for the TOE to physically protect itself from compromise of key material and user authentication data via physical access to the TOE as required by O.TAMPER. *FCS_COP.1.1.A* provides the capability for the TOE to encrypt the private keys and user passwords using DES. *FCS_CKM.4.1* provides the capability to delete the Master key by disconnection of battery as key is held in battery-backed volatile memory.

Table 18 – Informal Argument of Sufficiency

Given the arguments in Table 18 and the mappings shown in Table 17, it has been demonstrated that the security functional requirements are sufficient to enforce the security objectives for the TOE.

7.2.3 Rationale for EAL4 Assurance Level

In Part 3 of the CC EAL4 is defined as "methodically designed, tested and reviewed". This assurance level is therefore applicable in those circumstances where users require a methodically designed, tested, and reviewed product and also require a moderate to high level of independently assured security in conventional commodity security products and are prepared to incur additional security-specific engineering costs.

EAL4 assurance level has been chosen for the TOE as it is considered appropriate for the protection of sensitive information transmitted over public SONET/SDH, ATM, Ethernet and point-to-point data networks. It is also considered to be an appropriate level to counter the threats outlined in section 3 and satisfy the security objectives listed in section 4.

7.2.4 Strength of Function Claim

Users of the TOE may also access the CypherNET component by console port logon or remotely using SNMPv3. A user must present a User ID and an authentication password, which is then compared against the information contained in the user table. TSF F.IDENTIFICATION and SFR FIA_UAU.2.1are I&A probabilistic functions, therefore a strength of function claim is required for each function.

The SOF claim for F.IDENTIFICATION is basic.

The SOF claim for SFR FIA_UAU.2.1 is basic.

The SOF claim TOE is basic.

SOF-basic as defined in the CC Part 1 as a level of the TOE strength of function where analysis shows that the function provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential.

Access to the TOE requires a user name and password. The minimum password length is 8 characters. An attack trying all password combinations is not practical in a reasonable time and hence a SOF rating of basic can be claimed.

Further SFR FIA_UAU.2.1 maps directly to for F.IDENTIFICATION so an SOF claim for one applies to the other. Also as there are no other I&A probabilistic functions in the TOE, the SOF claim for the TOE is therefore dependent only on these functions.

There are a number of other security functions identified for the TOE where a strength of function claim is required. These strength of function claims are cryptographic functions. The National COMSEC authority determines the appropriateness of any cryptographic functions. Therefore, no strength of function claim has been provided for the following SFR's – FCS_CKM.1.A, FCS_CKM.1.B, FCS_CKM.1.C, FCS_CKM.2.A, FCS_COP.1.A, FCS_COP.1.B, FCS_COP.1.C, FCS_COP.1.F and FCS_COP.1.G.

7.2.5 SFR Dependencies Analysis

The table below shows those components that are dependent on other SFRs of the TOE.

Component:	Depends On:	Which Is:
FAU_GEN.1	FPT_STM.1	Included
FAU_SAR.1	FAU_GEN.1	Included
FCS_CKM.1.A	FCS_COP.1.A FCS_CKM.4 FMT_MSA.2.C	Included Included Included
FCS_CKM.1.B	FCS_COP.1.F FCS_CKM.4 FMT_MSA.2.C	Included Included Included
FCS_CKM.1.C	FCS_COP.1.A FCS_CKM.4 FMT_MSA.2.C	Included Included Included
FCS_CKM.2.A	FCS_CKM.1.C FCS_CKM.4 FMT_MSA.2.C	Included Included Included
FCS_CKM.4	FCS_CKM.1 FMT_MSA.2.C	Included Included
FCS_COP.1.A	FCS_CKM.1.A FCS_CKM.4 FMT_MSA.2.C	Included Included Included
FCS_COP.1.B	FCS_CKM.1.A FCS_CKM.4 FMT_MSA.2.C	Included Included Included
FCS_COP.1.C	FCS_CKM.1.A FCS_CKM.4 FMT_MSA.2.C	Included Included Included
FCS_COP.1.F	FCS_CKM.1.B FCS_CKM.4 FMT_MSA.2.C	Included Included Included
FCS_COP.1.G	FCS_CKM.1.A FCS_CKM.4 FMT_MSA.2.C	Included Included Included
FDP_ACC.1	FDP_ACF.1	Included

FDP_ACF.1	FDP_ACC.1 FMT_MSA.3.B	Included Included
FDP_DAU.1	—	No dependencies.
FDP_IFC.1	FDP_IFF.1	Included
FDP_IFF.1	FDP_IFC.1 FMT.MSA.3.A	Included Included
FDP_UCT.1	FTP_ITC.1 FDP_IFC.1	Included Included
FIA_AFL.1	FIA_UAU.1	Not included. FIA_UAU.2 used
FIA_UAU.2	FIA_UID.1	Not included. FIA_UID.2 used
FIA_UID.2	—	No dependencies
FMT_MSA.1.A	FDP_IFC.1 FMT_SMR.1	Included Included
FMT_MSA.1.B	FDP_ACC.1 FMT_SMR.1	Included Included
FMT_MSA.2.A	ADV_SPM.1 FDP_IFC.1 FMT_MSA.1.A FMT_SMR.1	Refer to TOE Informal Security Policy Model Included Included Included
FMT_MSA.2.B	ADV_SPM.1 FDP_ACC.1 FMT_MSA.1.B FMT_SMR.1	Refer to TOE Informal Security Policy Model Included Included Included
FMT_MSA.2.C	ADV_SPM.1 FDP_ACC.1 FMT_MSA.1 FMT_SMR.1	Included Included Included Included
FMT_MSA.3.A	FMT_MSA.1.A FMT_SMR.1	Included Included
FMT_MSA.3.B	FMT_MSA.1.B FMT_SMR.1	Included Included
FMT_MTD.1	FMT_SMR.1	Included
FMT_SMR.1	FMT_UID.1	Not included. FIA_UID.2 used.
FPT_AMT.1	—	No dependencies.

FPT_FLS.1	ADV_SPM.1	Refer to TOE Informal Security Policy Model
FPT_ITT.1	—	No dependencies.
FPT_PHP.3	—	No dependencies.
FPT_STM.1	—	No dependencies.
FPT_TST.1	FPT_AMT.1	Included
FTA_SSL.3	—	No dependencies.
FTP_ITC.1	—	No dependencies.

Table 19 – SFR Dependencies Analysis

7.2.6 Demonstration of Mutual Support

The analysis to demonstrate mutual support considers attacks in four main areas. Statements on how the TOE security functions mutually support each other in preventing these attacks are provided below:

Attack	Description
Tamper attacks are prevented by:	Security functions that restrict the modification of attributes or configuration data to authorised administrators and supervisors only (e.g. those based on *FMT_MSA.1*). Security functions that prevent the unauthorised modification of other data, the integrity of which is critical to a security function (i.e. those based on *FMT_MTD.1*). Security functions that protect the confidentiality of management information from the remote management host (CypherManager) to CypherNET (i.e. *FPT_ITT.1 and FCS_COP.1.1.A*). Security functions that protect against 'man-in-the-middle' attacks by authenticating data generated during X.509 certificate loading of the TOE (i.e. *FDP_DAU.1*].
Bypassing attacks are prevented by:	Security functions that require user identification and user authentication prior to allowing the user to perform actions on the security attributes of the TOE (based on *FIA_AFL.1, FTA_SSL.3, FIA_UAU.2, FIA_UID.2*). Security functions that allow for handling of SONET/SDH frames, ATM cells, Ethernet frames and bit streams on the basis of header information, line/path number or channel number. (i.e. *FDP_IFC.1 & FDP_IFF.1*).
Failure of Detection of Misconfiguration is prevented by:	Security functions that capture security relevant events (e.g. unsuccessful user identification and authentication) (based on *FAU_GEN.1*) Security functions that provide security relevant audit data in a readable form to allow for detection of misconfigurations (based on *FAU_SAR.1*) *FPT_STM.1.1* that provides reliable time stamping for each security relevant event. Abstract machine testing that allows for the determination of correct/incorrect operation of the abstract machine that underlies all other security functions (*FPT_AMT.1 & FPT_TST.1*). *FPT_PHP.3* that resists an attempt to gain physical access to the key material and password data, and takes automatic action.

Attack	Description
Breach of Transmitted Information Confidentiality is prevented by:	Security functions that provide for key generation, key distribution and key destruction (*FCS_CKM.1, FCS_CKM.2* and *FCS_CKM.4*). Security functions that allow only secure values are accepted for secure attributes (*FMT_MSA.2*).

Table 20 – Demonstration of Mutual Support

Table 19 shows those security functions that are dependent upon other security functions of the TOE. The informal analysis provided above, shows how those dependent security functions are mutually supportive in satisfying the security functional requirements of the TOE.

7.3 TOE Summary Specification Rationale

7.3.1 Mapping of Security Functional Requirements versus TOE IT Security Functions

The following table shows that each SFR is mapped to at least one TSF.

TOE IT Security Function Security Functional Requirement	F.AUDIT	F.CERTIFICATE_MANAGEMENT	F.DATA_EXCHANGE	F.IDENTIFICATION	F.KEY_MANAGEMENT	F.INFORMATION_FLOW_CONTROL	F.ROLE_BASED_ACCESS	F.SECURE_REMOTE_MANAGEMENT	F.SELF_PROTECT
FAU_GEN.1.1	✓								
FAU_GEN.1.2	✓								
FAU_SAR.1.1	✓								
FAU_SAR.1.2	✓								
FCS_CKM.1.1.A					✓				
FCS_CKM.1.1.B					✓				
FCS_CKM.1.1.C					✓				
FCS_CKM.2.1.A					✓				
FCS_CKM.4.1					✓				✓
FCS_COP.1.1.A			✓					✓	✓
FCS_COP.1.1.B			✓						
FCS_COP.1.1.C		✓			✓				
FCS_COP.1.1.F		✓							
FCS_COP.1.1.G		✓							
FDP_ACC.1.1							✓		
FDP_ACF.1.1							✓		
FDP_ACF.1.2							✓		
FDP_ACF.1.3							✓		
FDP_ACF.1.4							✓		
FDP_DAU.1.1		✓							
FDP_DAU.1.2		✓							
FDP_IFC.1.1						✓			
FDP_IFF.1.1						✓			
FDP_IFF.1.2						✓			
FDP_IFF.1.3						✓			

TOE IT Security Function **Security Functional Requirement**	**F.AUDIT**	**F.CERTIFICATE_** **MANAGEMENT**	**F.DATA_** **EXCHANGE**	**F.IDENTIFICATION**	**F.KEY_** **MANAGEMENT**	**F.INFORMATION_** **FLOW_CONTROL**	**F.ROLE_BASED_ACCESS**	**F.SECURE_REMOTE_MANAGEM** **ENT**	**F.SELF_PROTECT**
FDP_IFF.1.4						✓			
FDP_IFF.1.5						✓			
FDP_IFF.1.6						✓			
FDP_UCT.1.1			✓						
FIA_AFL.1.1				✓					
FIA_AFL.1.2				✓					
FIA_UAU.2.1				✓					
FIA_UID.2.1				✓					
FMT_MSA.1.1.A							✓		
FMT_MSA.1.1.B							✓		
FMT_MSA.2.1.A						✓			
FMT_MSA.2.1.B							✓		
FMT_MSA.2.1.C					✓				
FMT_MSA.3.1.A						✓			
FMT_MSA.3.1.B							✓		
FMT_MSA.3.2.A						✓			
FMT_MSA.3.2.B							✓		
FMT_MTD.1.1							✓		
FMT_SMR.1.1							✓		
FMT_SMR.1.2							✓		
FPT_AMT.1.1									✓
FPT_FLS.1.1									✓
FPT_ITT.1.1								✓	
FPT_PHP.3.1.A									✓
FPT_PHP.3.1.B									✓
FPT_STM.1.1	✓								
FPT_TST.1.1									✓
FPT_TST.1.2									✓
FPT_TST.1.3									✓
FTA_SSL.3.1							✓		

TOE IT Security Function **Security Functional Requirement**	**F.AUDIT**	**F.CERTIFICATE_ MANAGEMENT**	**F.DATA_ EXCHANGE**	**F.IDENTIFICATION**	**F.KEY_ MANAGEMENT**	**F.INFORMATION_ FLOW_CONTROL**	**F.ROLE_BASED_ACCESS**	**F.SECURE_REMOTE_MANAGEM ENT**	**F.SELF_PROTECT**
FTP_ITC.1.1		✓							
FTP_ITC.1.2		✓							
FTP_ITC.1.3		✓							

Table 21 – Mapping of SFRs to TOE Security Functions

7.3.2 Demonstration of Suitability

The following table provides an analysis of the TOE IT Security Functions and provides the justification for the suitability of the TSF's to satisfy the SFR's.

Security Functional Requirement	TOE Security Functions	Justification
FAU_GEN.1.1	F.AUDIT	*FAU_GEN.1.1* is completely met by *F.AUDIT* as this TSF generates an audit record for each of the auditable events.
FAU_GEN.1.2	F.AUDIT	*FAU_GEN.1.2* is completely met by *F.AUDIT* as this TSF generates an audit record with the required audit information for each audit event.
FAU_SAR.1.1	F.AUDIT	*FAU_SAR.1.1* is completely met by *F.AUDIT* as this TSF provides the ability for an authorised user to read the audit information from the audit records.
FAU_SAR.1.2	F.AUDIT	*FAU_SAR.1.2* is completely met by *F.AUDIT* as this TSF provides the ability to display audit information in a manner to enable a user to interpret the audit information.
FCS_CKM.1.1.A	F.KEY_ MANAGEMENT	*FCS_CKM.1.1.A* is completely met by *F.KEY_MANAGEMENT* as this TSF provides the functionality to generate keys in accordance with the specified standards.
FCS_CKM.1.1.B	F.KEY_ MANAGEMENT	*FCS_CKM.1.1.B* is completely met by *F.KEY_MANAGEMENT* as this TSF provides the functionality to generate keys in accordance with a specified standard.
FCS_CKM.1.1.C	F.KEY_ MANAGEMENT	*FCS_CKM.1.1.C* is completely met by *F.KEY_MANAGEMENT* as this TSF provides the functionality to generate keys in accordance with a specified standard.
FCS_CKM.2.1.A	F.KEY_ MANAGEMENT	*FCS_CKM.2.1.A* is completely met by *F.KEY_MANAGEMENT* as this TSF provides the functionality to distribute generate keys in accordance with a specified standard.

Security Functional Requirement	TOE Security Functions	Justification
FCS_CKM.4.1	F.KEY_ MANAGEMENT F.SELF_PROTECT	*FCS_CKM.4.1* is partially met by *F.KEY_MANAGEMENT* as this TSF provides the functionality to destroy all generated session keys. The session keys are held in volatile memory and are lost on power loss to the Encryptor. *FCS_CKM.4.1* is partially met by *F.SELF_PROTECT* as this TSF provides the functionality to destroy the Encryptor's Master key. The Master key is held in volatile battery-backed memory and will be lost on disconnection to the battery, which will occur if the Encryptor case is tampered and opened. The Master Key encrypts Critical Security Parameters, such as user passwords and the encryptor's RSA private key.
FCS_COP.1.1.A	F.DATA_ EXCHANGE F.SECURE_ REMOTE_ MANAGEMENT F.SELF_PROTECT	*FCS_COP.1.1.A* is met by three separate functions: one for encrypting user data payloads, and one for encrypting management data, and one for encrypting the CypherManager private key. *FCS_COP.1.1.A* is met by *F.DATA_EXCHANGE*, which provides cryptographic operation functionality to perform hardware-or software based 56, 112 or 168 bit DES encryption in CFB or counter mode on the cell, frame or packet payload or CFB mode on the bit stream. *FCS_COP.1.1.A* is met by *F.SECURE_REMOTE_MANAGEMENT* as the TSF provides the capability to encrypt all data transmitted between CypherManager and the CypherNet encryptor using DES algorithm. *FCS_COP.1.1.A* is met by *F.SELF_PROTECT* which encrypts its own private key on CypherManager using triple DES and a passphrase.

Security Functional Requirement	TOE Security Functions	Justification
FCS_COP.1.1.B	F.DATA_ EXCHANGE	*FCS_COP.1.1.B* is completely met by *F.DATA_EXCHANGE* which provides cryptographic operation functionality to perform 128 or 256 bit AES encryption in CFB or counter mode on the cell, frame or packet payload or CFB mode on the bit stream or 256 bit AES encryption on the SONET/SDH payload.
FCS_COP.1.1.C	F.KEY_ MANAGEMENT F.CERTIFICATE_ MANAGEMENT	*FCS_COP.1.1.C* is partially met by *F.KEY_MANAGEMENT* which provides key management for public keys within RSA cryptographic operations. *FCS_COP.1.1.C* is partially met by *F.CERTIFICATE_MANAGEMENT* which provides certificate management for public keys within RSA cryptographic operations. *FCS_COP.1.1.C* exclusively covers the RSA cryptographic operations specification which is covered by key management and certificate management covered in *F.KEY_MANAGEMENT* and *F.CERTIFICATE_MANAGEMENT.*
FCS_COP.1.1.F	F.CERTIFICATE_ MANAGEMENT	*FCS_COP.1.1.F* is completely met by *F.CERTIFICATE_MANAGEMENT* which provides Secure Hash Standard which is validates digital signatures as per FIPS PUB 180-1.
FCS_COP.1.1.G	F.CERTIFICATE_ MANAGEMENT	*FCS_COP.1.1.G* is completely met by *F.CERTIFICATE_MANAGEMENT,* which provides the RSA cryptographic functionality to generate digital signatures for X.509 certificates.

Security Functional Requirement	TOE Security Functions	Justification
FDP_ACC.1.1	F.ROLE_BASED_ ACCESS	*FDP_ACC.1.1* is completely met by *F.ROLE_BASED_ACCESS* as the TSF defines the subjects, objects and operations enforced by the Management Access Control SFP. The subjects are packets received by the encryptor either by SNMPv3 or data received on the local console. The objects are the encryptor information, and the operations are the actions that can be performed upon the objects. The operations are defined by the human user's role of either Administrator, Supervisor or Operator.
FDP_ACF.1.1	F.ROLE_BASED_ ACCESS	*FDP_ACF.1.1* is completely met by *F.ROLE_BASED_ACCESS* as the TSF provides the functionality to enforce the access control SFP to objects based on the user's ID and user's authentication password. Depending upon the interface used by the user, access is also governed by the privacy password (for SNMPv3).
FDP_ACF.1.2	F.ROLE_BASED_ ACCESS	*FDP_ACF.1.2* is completely met by *F.ROLE_BASED_ACCESS* as the TSF provides the rules that will control access to the encryptor's management operations.
FDP_ACF.1.3	F.ROLE_BASED_ ACCESS	*FDP_ACF.1.3* is completely met by *F.ROLE_BASED_ACCESS* as the TSF provides the rules that will control access to the encryptor's management operations.
FDP_ACF.1.4	F.ROLE_BASED_ ACCESS	*FDP_ACF.1.4* is completely met by *F.ROLE_BASED_ACCESS* as the TSF provides the functionality to deny access to the TOE if: • a trusted path cannot be established, or • the user id is not listed in the user table, or • the incorrect user password is provided, or • the data field in the SNMP v3 packet cannot be decrypted

Security Functional Requirement	TOE Security Functions	Justification
FDP_DAU.1.1	F.CERTIFICATE_ MANAGEMENT	*FDP_DAU.1.1* is completely met by *F.CERTIFICATE_MANAGEMENT* as the TSF generates a hash value that can be used as evidence to guarantee the validity of the X.509 certificate.
FDP_DAU.1.2	F.CERTIFICATE_ MANAGEMENT	*FDP_DAU.1.2* is completely met by *F.CERTIFICATE_MANAGEMENT* as the TSF displays the hash value that allows an administrator to verify the validity of the X.509 certificate.
FDP_IFC.1.1	F.INFORMATION_ FLOW_CONTROL	*FDP_IFC.1.1* is completely met by *F.INFORMATION_FLOW_CONTROL* as the TSF enforces the operation (encrypt, bypass or discard) on SONET/SDH frames, ATM cells, Ethernet frames and bit streams received on the local and network interfaces.
FDP_IFF.1.1	F.INFORMATION_ FLOW_CONTROL	*FDP_IFF.1.1* is completely met by *F.INFORMATION_FLOW_CONTROL* as the TSF enforces the operation (encrypt, bypass or discard) based information in the SONET/SDH frames, ATM cells and Ethernet frames or channel number for bit stream connections received on the local and network interfaces.
FDP_IFF.1.2	F.INFORMATION_ FLOW_CONTROL	*FDP_IFF.1.2* is completely met by *F.INFORMATION_FLOW_CONTROL* as the TSF enforces the rules as specified in the CAT on SONET/SDH frames, ATM cells Ethernet frames and bit streams received on the local and network interfaces.
FDP_IFF.1.3	F.INFORMATION_ FLOW_CONTROL	*FDP_IFF.1.3* is completely met by *F.INFORMATION_FLOW_CONTROL* as the TSF enforces “encrypt”, “bypass” or “discard” action as specified in the CAT on SONET/SDH frames, ATM cells Ethernet frames and bit streams received on the local and network interfaces.

Security Functional Requirement	TOE Security Functions	Justification
FDP_IFF.1.4	F.INFORMATION_ FLOW_CONTROL	*FDP_IFF.1.4* is completely met by *F.INFORMATION_FLOW_CONTROL* as the TSF enforces no additional the rules on SONET/SDH frames, ATM cells, Ethernet frames and bit streams received on the local and network interfaces.
FDP_IFF.1.5	F.INFORMATION_ FLOW_CONTROL	*FDP_IFF.1.5* is completely met by *F.INFORMATION_FLOW_CONTROL* as the TSF enforces the rules as specified by the CAT table on SONET/SDH frames, ATM cells, Ethernet frames and bit streams received on the local and network interfaces.
FDP_IFF.1.6	F.INFORMATION_ FLOW_CONTROL	*FDP_IFF.1.6* is completely met by *F.INFORMATION_FLOW_CONTROL* as the TSF denies information flow if a connection for SONET/SDH frames, ATM cells, Ethernet frames and bit streams received on the local and network interfaces is not listed in the CAT table.
FDP_UCT.1.1	F.DATA_ EXCHANGE	*FDP_UCT.1.1* is completely met by *F.DATA_EXCHANGE* as the TSF enables information to be transmitted and received, to be encrypted, which protects the information from unauthorised disclosure.
FIA_AFL.1.1	F.IDENTIFICATION	*FIA_AFL.1.1* is completely met by *F.IDENTIFICATION* as the TSF ensures that three unsuccessful attempts are detected.
FIA_AFL.1.2	F.IDENTIFICATION	*FIA_AFL.1.2* is completely met by *F.IDENTIFICATION* as the TSF ensures that after three unsuccessful attempts to gain access to the TOE are detected the user account is disabled for three minutes.
FIA_UAU.2.1	F.IDENTIFICATION	*FIA_UAU.2.1* is completely met by *F.IDENTIFICATION* as the TSF does not allow access to the TOE unless the user has successfully been authenticated.

Security Functional Requirement	TOE Security Functions	Justification
FIA_UID.2.1	F.IDENTIFICATION	*FIA_UID.2.1* is completely met by *F.IDENTIFICATION* as the TSF requires each user to be successfully identified before any access to the TOE is allowed.
FMT_MSA.1.1.A	F.ROLE_BASED_ ACCESS	*FMT_MSA.1.1.A* is completely met by *F.ROLE_BASED_ACCESS* as the TSF enforces Management Access Control SFP to restrict the ability to make changes to the security attributes to only authorised Administrators and Supervisors.
FMT_MSA.1.1.B	F.ROLE_BASED_ ACCESS	*FMT_MSA.1.1.B* is completely met by *F.ROLE_BASED_ACCESS* as the TSF enforces Management Access Control SFP to restrict the ability to make changes to the security attributes to authorised users with the appropriate role.
FMT_MSA.2.1.A	F.INFORMATION_ FLOW_CONTROL	*FMT_MSA.2.1.A* is completely met by *F.INFORMATION_FLOW_CONTROL* as the TSF ensures that only secure values are accepted for security attributes. The security attributes are the address details and the action (encrypt, bypass, discard) that will determine how the information flow is treated. This TSF will enforce the policy as configured in the CAT and there are no security attribute settings that will result in an insecure state. Note a “Bypass” policy is not considered insecure as the TOE is actioning the configured policy.

Security Functional Requirement	TOE Security Functions	Justification
FMT_MSA.2.1.B	F.ROLE_BASED_ ACCESS	*FMT_MSA.2.1.B* is completely met by *F.ROLE_BASED_ACCESS* as the TSF ensures that only secure values are accepted for security attributes. The security attributes and secure states are: • User accounts: the password(s) must meet lexical complexity and uphold a secure state. An empty or weak password would result in an insecure state. • X.509 Certificates: must be signed by same CA for Encryptors to authenticate and establish a secure session. If one Encryptor has a certificate from a different CA then authentication will fail and no traffic will be transmitted and so upholding security. • Firmware: images are encrypted by Senetas and decrypted by the Encryptor so only trusted images can be loaded onto the Encryptor.
FMT_MSA.2.1.C	F.KEY_ MANAGEMENT	*FMT_MSA.2.1.C* is completely met by *F.KEY_MANAGEMENT* as the TSF ensures that only secure cryptographic key values are accepted for security attributes
FMT_MSA.3.1.A	F.INFORMATION_ FLOW_CONTROL	*FMT_MSA.3.1.A* is completely met by *F.INFORMATION_FLOW_CONTROL* as the TSF ensures that restrictive default values are set. The CAT table initially contains no entries hence all received information on the local and network ports is discarded.
FMT_MSA.3.1.B	F.ROLE_BASED_ ACCESS	*FMT_MSA.3.1.A* is completely met by *F.ROLE_BASED_ACCESS* as the TSF ensures that restrictive default values are set. The User table initially has one default administrator account. All other users are created as operators unless the administrator overrides this value

Security Functional Requirement	TOE Security Functions	Justification
FMT_MSA.3.2.A	F.INFORMATION_FLOW_CONTROL	*FMT_MSA.3.2.A* is completely met by *F.INFORMATION_FLOW_CONTROL* as the TSF ensures that restrictive default values are set.
FMT_MSA.3.2.B	F.ROLE_BASED_ACCESS	*FMT_MSA.3.2.B* is completely met by *F.ROLE_BASED_ACCESS* as the TSF ensures that only administrators and supervisors can change the default values.
FMT_MTD.1.1	F.ROLE_BASED_ACCESS	*FMT_MTD.1.1* is completely met by *F.ROLE_BASED_ACCESS* as the defined roles (Administrator, Supervisor, Operator) will have access to manage TSF data as their role allows.
FMT_SMR.1.1	F.ROLE_BASED_ACCESS	*FMT_SMR.1.1* is completely met by *F.ROLE_BASED_ACCESS* as the TSF maintains the roles of administrator, supervisor and operator.
FMT_SMR.1.2	F.ROLE_BASED_ACCESS	*FMT_SMR.1.1* is completely met by *F.ROLE_BASED_ACCESS* as the TSF enables each user to be associated with a role.
FPT_AMT.1.1	F.SELF_PROTECT	*FPT_AMT.1.1* is completely met by *F.SELF_PROTECT* as the TSF provides the ability to run tests to ensure the correct operation of the TOE.
FPT_FLS.1.1	F.SELF_PROTECT	*FPT_FLS.1.1* is completely met by *F.SELF_PROTECT* as the TSF ensures that if a self-test fails the TOE will enter a secure halt mode preventing transmission of any information.
FPT_ITT.1.1	F.SECURE_REMOTE_MANAGEMENT	*FPT_ITT.1.1* is completely met by *F.SECURE_REMOTE_MANAGEMENT* as the TSF encrypts all data transmitted between physical separate parts of the TOE
FPT_PHP.3.1.A	F.SELF_PROTECT	*FPT_PHP.3.1.A* is completely met by *F.SELF_PROTECT* as the TSF ensures the Master key protecting private key material is automatically erased if the TOE is tampered. Without the Master key the private key material cannot be accessed.

Security Functional Requirement	TOE Security Functions	Justification
FPT_PHP.3.1.B	F.SELF_ PROTECT	*FPT_PHP.3.1.B* is completely met by *F.SELF_PROTECT* as the TSF ensures the Master key protecting all user passwords is automatically erased if the TOE is tampered. Without the Master key the user passwords cannot be accessed.
FPT_STM.1.1	F.AUDIT	FPT_STM.1.1 is met by *F.AUDIT*, as this TSF provides a real time clock to timestamp audit records.
FPT_TST.1.1	F.SELF_ PROTECT	*FPT_TST.1.1* is completely met by *F.SELF_PROTECT* as the TSF ensures that self-tests are run at power up
FPT_TST.1.2	F.SELF_ PROTECT	*FPT_TST.1.2* is completely met by *F.SELF_PROTECT* as the TSF allows authorised users to verify the integrity of TSF configuration data.
FPT_TST.1.3	F.SELF_ PROTECT	*FPT_TST.1.3* is completely met by *F.SELF_PROTECT* as the TSF allows authorised users to verify the integrity of the stored executable code.
FTA_SSL.3.1	F.ROLE_BASED_ACCESS	*FTA_SSL.3.1* is completely met by *F.ROLE_BASED_ACCESS* as the TSF provides the functionality to terminate an interactive session after a pre-defined time period.
FTP_ITC.1.1	F.CERTIFICATE_ MANAGEMENT	*FTP_ITC.1.1* is completely met by *F.CERTIFICATE_MANAGEMENT* as the TSF provides for the Encryptor to establish a secure communications channel with other Encryptors that have X.509 certificates signed by the same CA to assure identity of the end-points. The data exchanged between the two Encryptors is then protected from modification or disclosure.

Security Functional Requirement	TOE Security Functions	Justification
FTP_ITC.1.2	F.CERTIFICATE_ MANAGEMENT	*FTP_ITC.12* is completely met by *F.CERTIFICATE_MANAGEMENT* as the TSF provides the functionality to permit either Encryptor (the TSF or the remote trusted IT product) to initiate communication via the trusted channel.
FTP_ITC.1.3	F.CERTIFICATE_ MANAGEMENT	*FTP_ITC.1.3* is completely met by *F.CERTIFICATE_MANAGEMENT* as the TSF provides the trusted channel through which the bulk data session keys can be exchanged that are used for enforcing the Information Flow Control SFP.

Table 22 – Demonstration of Suitability

From the mappings provided in Table 21 it can be clearly seen that all TSFs map to at least one SFR and also that all SFRs map to at least one TSF. Given the arguments above and the complete mappings shown in Table 21, and the fact that all SFR’s were derived from the Common Criteria Part 2: Security Functional Requirements, it can be concluded that the TOE security functions are suitable to meet the security functional requirements for the TOE.

7.3.3 Demonstration of Mutual Support

It is not necessary to repeat an analysis of TOE Security Function binding at this level given that:

a. All functions at this level were derived from the Security Functional Requirements identified in section 5.1of the Security Target; and

b. In section 7.2 all security functions were shown to bind together to provide a mutually supportive and effective whole.

Therefore, it is possible to conclude that all TOE Security Functions identified at this level bind together to provide a mutually supportive and effective whole.

7.3.4 Assurance Requirements Rationale

All SAR’s have been included, as specified in the CC part 3, for an EAL4 assurance level. The requirements for EAL4 level of assurance were justified in section 7.2.3 of this Security Target and the assurance measures are listed in Table 12 – TOE Assurance Measures. Hence the SAR’s listed in Table 12 provide the necessary assurance for an EAL4 evaluation.

End