ITT INDUSTRIES
DRAGONFLY GUARD
SECURITY TARGET
VERSION 2.0

Kristina C. Rogers

29 October 1998

CYGNACOM SOLUTIONS

Suite 100 West♦7927 Jones Branch Drive♦McLean, VA 22102-3305♦703 848-0883♦Fax 703 848-0960

TABLE OF CONTENTS

SECTION **PAGE**

TABLE OF TABLES

TABLE **PAGE**

1.0 SECURITY TARGET INTRODUCTION

1.1 SECURITY TARGET IDENTIFICATION

TOE Identification: ITT Industries Dragonfly Guard, Version 3.0, Build 980908.1509

ST Identification: ITT Industries Dragonfly Guard Security Target, Version 2.0

Assurance level: EAL2

Registration: <To be filled in upon registration>

Keywords: Guard, Firewall, In-Line Encryption, Network Security, Multilevel Security, Access Control, Tactical, Fortezza, Security Target

1.2 SECURITY TARGET OVERVIEW

The ITT Dragonfly Guard is a network security device that uses National Security Agency (NSA) Fortezza Cards to provide multi-level secure (MLS) services to legacy networks in System High mode. The Dragonfly Guard operates on standard Internet Protocol (IP) datagrams. The Dragonfly Guard provides the following security services: mandatory access control, discretionary access control, confidentiality, integrity, source authentication, and audit. The Dragonfly Guard cryptographically labels every IP Datagram with an appropriate security level, and then checks that label before releasing the underlying datagram in plain text form. The Dragonfly Guard provides discretionary access control between the domains that it protects. All User Data is encrypted and integrity checks are applied to all messages transmitted between two Dragonfly Guards. In order to provide these services, Dragonfly Guards set up a trusted Association based on source authentication and use the Fortezza Key Exchange Algorithm to generate a symmetric key. Any Dragonfly Guard can also be designated as an Audit Catcher. Audit Catchers receive audit reports from other Dragonfly Guards and send all messages to their serial port for printing, storage or subsequent analysis. The selection of auditable events can be dynamically controlled by updating an Audit Mask. Besides providing security services, the Dragonfly Guard offers configuration options that allow it to operate in a changing tactical environment where not all hosts are Dragonfly equipped.

1.3 COMMON CRITERIA CONFORMANCE

The Dragonfly Guard is Part 2 Conformant and Part 3 Conformant.

2.0 TOE DESCRIPTION

2.1 PRODUCT DESCRIPTION

The ITT Dragonfly Guard is a network security device that uses National Security Agency (NSA) Fortezza Cards to provide multi-level secure (MLS) services to legacy networks, i.e., Internet Protocol (IP) networks that operate in System High mode. The Dragonfly Guard can also serve as a firewall or an in-line encryptor. Dragonfly Guards protect enclaves or individual hosts. Within a network, Dragonfly Guards are in-line between the host and the network. Dragonfly Guards operate on standard IP datagrams.

A Dragonfly Guard is an enclosed unit containing a 486 motherboard and two Ethernet processors. The unit has two Personal Computer Memory Card International Association (PCMCIA) card slots, two Ethernet ports labeled local and remote, and a serial port.

Dragonfly Guards require two PCMCIA cards to operate. The first card is the Ignition Card that contains the Dragonfly software and is digitally signed. The second card is the User Fortezza Card that contains the configuration information for that particular Dragonfly Guard. The User Fortezza Card contains eight certificates. Five of them, the User, Configuration, Audit, the Certificate Revocation, and the Routing certificates, contain configuration information and are signed by the local authority. The other three are the local authority, the root, and the root authority certificates. The Dragonfly Guard uses the Fortezza card for hashing, digital signatures, key generation, and encryption.

Dragonfly Guards separate two Dragonfly Domains. A Dragonfly Domain is a set of computers that are networked together without any intervening Dragonfly Guards. These computers in the same domain may be PCs, Workstations, or Servers that are all at the same security level. The two domains are labeled remote and local for convenience, although processing is actually the same whether the port is local or remote.

The Dragonfly Administration System is used to define Dragonfly Domains and their properties. Initially, there is one Dragonfly Domain. The first Dragonfly Guard defined creates two domains: the Local Domain and the Remote Domain. When more than one Dragonfly Guard is connected to the same Dragonfly Domain, all of them must be configured with the same security level for that domain.

The Dragonfly Administration System is used to set the security and network configuration information. It then burns the information onto the User Fortezza Card for the Guard. The Administration System requires a Local Authority Fortezza Card to create valid User Fortezza Cards. The Local Authority Card is provided by ITT. The Administration System uses a graphical display and wizards to assist in the organization of Dragonfly Deployment, a set of Dragonfly Domains. As Dragonfly Guards are added to a Dragonfly Deployment, the security parameters for an existing domain are set from a Guard already protecting that Domain. The Dragonfly Guard depends upon the Dragonfly Administration System to correctly configure its User Fortezza Card. The configuration can be verified by checking the output of the serial port at initial start-up. The Dragonfly Administration System is outside the scope of this evaluation and is considered part of the environment for the Dragonfly Guard.

The Dragonfly Companion is a software product that resides on a host PC. The Dragonfly Companion provides the same security services as a Dragonfly Guard and is interoperable with it. Dragonfly Guards and Dragonfly Companions are collectively referred to as Dragonfly Units in Dragonfly documentation, but Dragonfly Guard and Dragonfly Unit are synonymous in this document. The Dragonfly Companion is being separately evaluated.

2.2 SECURITY SERVICES

The ITT Dragonfly Guard provides the following security services: source authentication, mandatory access control, discretionary access control, confidentiality, integrity, and audit.

Dragonfly Guards establish associations to authenticate each other, exchange security parameters, and establish a trusted session for communication.   Dragonfly Guards use the Fortezza card to generate and securely exchange a symmetric encryption key.

Dragonfly Guards always authenticate themselves to each other. All Dragonfly Messages sent before an association is formed or outside of an Association are digitally signed. This includes Association Requests and Association Grants.   After an association if formed, messages are encrypted with a symmetric key known only to the source and destination Dragonfly Guard. From a security policy perspective, the user on the Dragonfly Guard is the Dragonfly Guard itself. Dragonfly Guards identify and authenticate themselves to each other based on the identity associated with the User Certificate on their User Fortezza Card. The only role assumed by the Dragonfly Guard is the User Role. The Dragonfly Guard assumes the User Role when it logs into the User Fortezza Card using the PIN for the User certificate during initialization. The only direct human interface to the Dragonfly Guard is for the person who is responsible for connecting the remote and local ports, the serial port and the power supply and inserting the Ignition Card and the User Fortezza Card in the PCMCIA slots. To avoid confusion this person is called the installer, rather than the user in this document. Another possible area of confusion is the use of the term “User Data” . This term refers to data being sent on behalf of users on hosts in the data portions of IP datagrams. These users are not the same users as the Dragonfly Guard user. However, the term has not been modified, since User Data protection is a basic Common Criteria concept.

The Dragonfly Guard supports Mandatory Access Control (MAC) by labeling every IP Datagram with an appropriate security level, and then checking that label against the security level of the destination domain before releasing the underlying datagram in plain text form. Through the sharing of security related information via an Association, Dragonfly Guards can support both Write Equal and Write Up. In the Write Equal environment, where Dragonfly Domains are at the same security level, all IP based communications are allowed according to the MAC policy. Dragonfly also allows transfer of User Data from a low level Domain to a high level Domain called Write Up. In the case of Write Up, Dragonfly supports only the subset of IP based functionality for which the Dragonfly Guard can predict the response.

Many IP-based protocols require some form of feedback. For example, the file transfer protocol (FTP) uses flow control. The feedback constitutes a potential Write Down. Dragonfly assures that this Write Down does not constitute a violation of the security policy by a patented scheme of anticipated messages. Each feedback message is predicted by the Dragonfly Guard based upon the Write Up FTP or Simple Mail Transfer Protocol (SMTP) command. If the actual message matches the predicted message, the predicted message is released. Otherwise, no message is released and there is no feedback.

The Dragonfly Guard uses Privilege Vectors for Discretionary Access Control (DAC) between Dragonfly Domains. All communication allowed by DAC is bi-directional. Therefore, if the Privilege Vector of one domain allows communication with another, either Domain can initiate that communication. The primary advantage of this feature is that new domains can be added to a Deployment without requiring that the Privilege Vectors of existing Domains be updated. Access between existing domains and a new Domain can be allowed by the Privilege Vector of the new Domain. DAC checks are performed at the time an Association is formed.

The Dragonfly Guard provides Confidentiality of User Data. It uses a symmetric key generated using the Fortezza card to encrypt all User Data when it is transmitted between two Dragonfly Guards. The Guard uses the Cipher-Block Chaining CBC-64 mode of operation and the Skipjack algorithm on the User Fortezza Card.

The Dragonfly Guard checks for integrity of both User Data and Dragonfly control information when messages are transmitted between two Dragonfly Guards. Messages sent outside of an association are digitally signed. When a message is sent within an association, a checksum is computed and stored in the message before the message is encrypted.

Any Dragonfly Guard can also serve as an Audit Catcher. Audit Catchers receive audit reports from other Dragonfly Guards and send all messages to their serial port for printing, storage or subsequent analysis. The selection of auditable events can be controlled by updating an Audit Mask.

2.3 OPERATIONAL ENVIRONMENT

Besides providing security services, the Dragonfly Guard offers configuration options that allow it to operate in environments that dynamically change or where not all hosts are Dragonfly equipped.

Dragonfly Guards do not have to be programmed with complete deployment information as they use a trusted, automatic discovery mechanism to learn the system topology. Dragonfly Guards use Internet Control Message Protocol (ICMP) messages, ICMP Echo Requests (pings) and ICMP Echo Responses, to find out in which Dragonfly Domain a destination host is located. The ICMP Echo Request is transmitted at the same time as an Association Request. Once the Dragonfly Domain of the host is located, the source and destination Dragonfly Guards can exchange security levels and generate a symmetric key for encryption. Neither the initiating Dragonfly Guard nor the destination Dragonfly Guard needs to know the name, address, or even of the existence of the other prior to the Association setup. Once the association is set up, both Dragonfly Guards know all that they need to know.

The Dragonfly Guard supports mixed enclaves; i.e., where only a subset of hosts are equipped with Dragonfly Guards. This permits an organization to evolve to increasing degrees of security without requiring host platform upgrades or modification of existing applications. Mixed enclaves provide the capability to exchange data with a non-Dragonfly-protected host at the same security level. Non-Dragonfly-protected hosts are called Native hosts. Confidentiality and integrity protection is not provided as there is only one Dragonfly Guard along the data path and no encryption takes place.

The Dragonfly Guard provides in-line encryption (INE) functionality to tunnel data through a network at a different security level. Dragonfly Guards allow hosts at a lower security level to send communications through a network at a higher security level to another host at the same lower security level as the original host. Higher level information is not released to the lower level hosts. For example, two hosts at the SBU level could tunnel data through a Secret network. Also, hosts at a higher security level can communicate over a network at a lower security level without releasing information from the higher security level to the lower security level. For example, two hosts at the Sensitive but Unclassified (SBU) level could tunnel data through an unprotected Unclassified network. When two or more Dragonfly Guards exist along a data path, they provide confidentiality, integrity, and source authentication.

Each Dragonfly Guard can support multiple host devices connected to a LAN. This sharing of a Guard among a set of hosts (all operating at the same security level) allows a dramatic reduction in cost per host.

The Dragonfly Guard is designed to operate in a tactical environment where there is a requirement for a remote unclassified host to be able to transmit unclassified logistics data to an unclassified host on a secret tactical network. This can be done by putting Dragonfly Guards in front of the unclassified hosts and the secret network and using tunneling to transmit the data. However, in order for the Dragonfly Guard to operate on some tactical networks that configure themselves dynamically, the Guard must be able to identify itself to the internet gateway by means of Address Resolution Protocol (ARP)/ Reverse Address Resolution Protocol (RARP) requests/responses and the name server using the Name Server Requests/Responses. The Guard can be configured to allow or not allow ARP/RARP processing. If the Guard is configured to allow ARP/RARP processing , the Guard provides the capability of restricting these requests to specific hosts. (Writeups of Name Server Requests can be disabled only by disabling all writeups.)

3.0 SECURITY ENVIRONMENT

This section identifies the following:

- Secure usage assumptions,
- Organizational security policies, and
- Threats to Security

3.1 SECURE USAGE ASSUMPTIONS

Table 3.1 lists the Secure Usage Assumptions.

Assumption Name	Assumption Description
A. ADMIN	The local authority is trusted to correctly configure User Fortezza Cards.
A.ATTACK_LEVEL	Attackers are assumed to have a medium level of expertise, resources, and motivation.
A.CRYPTO_SERVICES	Cryptographic services are provided by the User Fortezza Card.
A.CRYPTO_SOF	The cryptographic algorithms on the Fortezza card are assumed to be strong enough to counter at least a medium level of attack.
A.ONLY_PATH	The Guard is assumed to be on the only data path between the two networks connected to its two Ethernet ports.
A.PHYSICAL	The Dragonfly Guard is assumed to be protected from physical tampering.
A.INSTALLER	Authorized installers are assumed to be able to insert the correct User Fortezza Card into the Dragonfly Guard and to connect the correct networks to the local and remote ports.

Table 3.1 – Secure Usage Assumptions

3.2 ORGANIZATIONAL SECURITY POLICIES

Table 3.2 lists the organizational security policies.

Policy Name	**Organizational Security Policy**
P.AUDIT	It must be possible to record security relevant actions.
P.DAC	It must be possible to control access between domains at the same security level.
P.MAC	A mandatory access control policy based on hierarchical security levels must be enforced. Information must not be allowed to flow from a higher security level to a lower security level.

Table 3.2 - Organizational Security Policies

3.3 THREATS TO SECURITY

Table 3.3 lists the threats to security.

No	Threat Name	Threat Description
1	T.Account	An attempted violation of the TSP may not be traceable to the Guard where it occurred.
2	T.Acquire_Key	An unauthorized user is able to acquire the key for an encrypted message.
3	T.Bypass	A user is able to bypass the security enforcing functions
4	T.Card_Lost	A Dragonfly Guard and its associated User Fortezza Card are lost and recovered by a malicious user.
5	T.Confidential	Data is released in violation of the TSP due to lack of confidentiality during transmission across an unprotected network.
6	T.Excess_Audit	It may not be possible to effectively analyze audit data due to an excessive volume of audit data being recorded.
7	T.Expired	A malicious user is able to use an old User Fortezza Card or an old cryptographic key to gain unauthorized access to information.
8	T.Hardware_Failure	The Dragonfly Guard performs incorrectly due to a hardware failure.
9	T.Impersonate	An unauthorized user may attempt to impersonate a Dragonfly Guard.
10	T.Inconsistent	An incorrect access control decision is made due to a security attribute being interpreted differently on another Dragonfly Guard.
11	T.Modify_Configuration	The Dragonfly Guard performs incorrectly due to either accidental or intentional modification of its configuration data.
12	T.Modify_Data	A message containing User or TSF Data may be modified during transmission.
13	T.Modify_Software	The Dragonfly Guard performs incorrectly due to either accidental or intentional modification of its software.
14	T.No_Need_To_Know	Users have access to data that they have no need to know.
15	T.Quit	A person changes or leaves a job.
16	T.Sequence	It may not be possible to determine the sequence of security relevant events.
17	T.Static_Audit	It may not be possible to record all the security relevant events when suspicious activity is observed due to an inability to dynamically change the set of events that are audited

No	Threat Name	Threat Description
18	T.Tamper	A malicious user is able to interfere with the execution of the TSF software or to modify internal TSF data.
19	T.Undetected	The occurrence of a suspicious security relevant event may go undetected due to the inability to record security relevant events.
20	T.Write_Down	Information at a higher security level is released on a network at a lower security level.
21	T.Wrong_Level	Exported or imported data may not be properly protected due to the TSF's inability to correctly associate a security level with data on export or import.

Table 3.3 – Threats to Security

4.0 SECURITY OBJECTIVES

4.1 SECURITY OBJECTIVES FOR THE TOE

Table 4.1 lists the security objectives for the TOE.

No	Objective Name	Objective Description
1	O.Accountability	A Guard collecting audit data must associate the security relevant events with the identity of the Guard from which they are reported.
2	O.Audit	The Guard must provide an audit capability that can record attempts to bypass the TOE Security Policy.
3	O.Audit_Select	The Guard must be able to change the selection of auditable events during normal operation.
4	O.Authen_Source	A Guard must authenticate itself to another Guard.
5	O.Confidentiality	User Data must be protected from disclosure when it is transmitted between two Guards.
6	O.Consistency	TSF Data must be interpreted consistently by all the Guards within a network.
7	O.DAC	The Guard must not release User Data to an unauthorized domain.
8	O.Domain_Separation	The Guard must maintain its own domain for execution and ensure that it cannot be interfered with or tampered with by an untrusted subject.
9	O.Expire	The Guard must provide for the expiration of user certificates and keys.
10	O.Info_Flow	The Guard must not release User Data from a higher level domain to a lower level domain.
11	O.Integrity	User Data and TSF Data must be protected from modification when it is transmitted between two Guards. A Guard must verify the integrity of User Data and TSF data when it is received.
12	O.Non-Bypassability	The Guard must ensure that a packet cannot be released until the security enforcing functions have been invoked and succeed.
13	O.Revoke	The Guard must provide for the revocation of user certificates.
14	O.Self_Test	The Guard must provide and execute self tests during initial start-up to ensure the integrity of its hardware and software.
15	O.Single_Level_Port	The Guard must assume that all hosts within a Dragonfly Domain are at the same level as the port to which they are connected.
16	O.SOF	The Guard must be able to meet at least a medium strength of function requirement.

No	Objective Name	Objective Description
17	O.Time	It must be possible to determine the time of security relevant events.
18	O.Trusted_Channel	Guards must be able to establish a trusted communication channel between each other.
19	O.Verify_Config	A Guard must be able to verify that its configuration certificates have been signed by the local authority.

Table 4.1 – Security Objectives for the TOE

4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT

Table 4.2 lists IT Security Objectives for the environment.[1]

No.	Objective Name	Objective Description
3E	O_E.Audit_Select	The Guard must be able to change the selection of auditable events during normal operation.
4E	O_E.Authen_Source	A Guard must authenticate itself to another Guard.
5E	O_E.Confidentiality	User Data must be protected from disclosure when it is transmitted between two Guards.
7E	O_E.DAC	The Guard must not release User Data to an unauthorized domain.
9E	O_E.Expire	The Guard must provide for the expiration of user certificates and keys.
10E	O_E.Info_Flow	The Guard must not release User Data from a higher level domain to a lower level domain.
11E	O_E.Integrity	User Data and TSF Data must be protected from modification when it is transmitted between two Guards. A Guard must verify the integrity of User Data and TSF data when it is received.
13E	O_E.Revoke	The Guard must provide for the revocation of user certificates.
15E	O_E.Single_Level_Port	The Guard must assume that all hosts within a Dragonfly Domain are at the same level as the port to which they are connected.
16E	O_E.SOF	The Guard must be able to meet at least a medium strength of function requirement.
18E	O_E.Trusted_Channel	Guards must be able to establish a trusted communication channel between each other.
19E	O_E.Verify_Config	A Guard must be able to verify that its configuration certificates have been signed by the local authority.

Table 4.2 – IT Security Objectives for the Environment

[1] Note that many of the Security Objectives for the TOE are also partially satisfied by the environment

Table 4.3 lists Non-IT Security Objectives for the environment.

No.	Objective Name	Objective Description
20	O-NON-IT.ADMIN	The local authority must be adequately trained on how to configure the User Fortezza Card.
21	O-NON-IT.ONLY_PATH	The Dragonfly Guard must be the only data path between the two networks that it is separating.
22	O-NON-IT.PHYSICAL	The Dragonfly Guard must be protected from physical tampering.
23	O-NON-IT.INSTALLER	The Dragonfly Guard Installer must be adequately trained on connecting the Ethernet ports and inserting the correct User Fortezza Card.

Table 4.3 – Non-IT Security Objectives for the Environment

5.0 IT SECURITY REQUIREMENTS

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS

This section contains the security functional requirements for the TOE. All of the functional requirements have been taken from Part 2 of the Common Criteria and none of them have been refined. The functional components are listed in Table 5.1.

No.	Component	Component Name
Class FAU: Audit		
1	FAU_GEN.1	Audit data generation
2	FAU_SEL.1	Selective audit
Class FDP: User Data Protection		
3	FDP_ACC.1	Subset access control
4	FDP_ACF.1	Security attribute based access control
5	FDP_ETC.1	Export of user data without security attributes
6	FDP_IFC.1	Subset information flow control
7	FDP_IFF.2	Hierarchical security attributes
8	FDP_ITC.1	Import of user data without security attributes
9	FDP_UCT.1	Basic data exchange confidentiality
10	FDP_UIT.1	Data exchange integrity
Class FIA: Identification and Authentication		
11	FIA_ATD.1	User attribute definition
12	FIA_UAU.2	User authentication before any action
13	FIA_UID.2	User identification before any action
Class FMT: Security Management		
14	FMT_MTD.1	Management of TSF Data
15	FMT_REV.1	Revocation
16	FMT_SAE.1	Time-limited authorisation
17	FMT_SMR.1	Security roles
Class FPT: Protection of the TOE Security Functions		
18	FPT_AMT.1	Abstract Machine Testing
19	FPT_ITI.1	Inter-TSF detection of modification
20	FPT_RVM.1	Non-bypassability of the TSP
21	FPT_SEP.1	TSF domain separation
22	FPT_STM.1	Reliable time stamps
23	FPT_TDC.1	Inter-TSF basic TSF data consistency
Class FTP: Trusted Path/Channels		
24	FTP_ITC.1	Inter-TSF Trusted Channel

Table 5.1 – Functional Components

The following sections contain the functional components from the Common Criteria (CC) Part 2 with the operations completed. The standard CC text is in regular font; the text inserted by the Security Target (ST) author is in italic font enclosed in brackets.

5.1.1 Class FAU: Security audit

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 [*not specified*] level of audit; and
c) [*Closing a Write Up,*
d) Anticipated Message Mismatch,
e) Anticipated Message Not allowed,
f) Anticipated Message Unknown,
g) Association Request Denied (Reported by Responder),
h) Association Request Denied (Reported by Initiator),
i) Association Closed,
j) Association Granted,
k) Association Requested,
l) Association Unknown,
m) Audit Mask Received,
n) Opening a Write Up Session,
o) Certificate or Symmetric Key Deleted,
p) Invalid Signature,
q) Lost Wait Queue Msg,
r) Received by non-Audit Catcher,
s) Certificate Revocation List Sent,
t) Old CRL Version,
u) Certificate Invalid Start,
v) Certification Expired,
w) Certificate Revoked,
x) Certificate Invalid, and
y) Security Level Mismatch.]

Note: If a Guard is configured to generate audit event messages, it will never generate an audit record of the shutdown of the audit functions.   The Guard has no interface to shutdown auditing without shutting down the Guard.

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*none].*

Dependencies: FPT_STM.1 Reliable time stamps

FAU_SEL.1 Selective audit

Hierarchical to: No other components.

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attribute: [*event type*].

Dependencies: FAU_GEN.1 Audit data generation

FMT_MTD.1 Management of TSF data

ITENV.3 Dragonfly Administration System for Setting User Attributes

ITENV.4 Dragonfly Administration System for Modifying TSF Data

5.1.2 Class FDP: User data protection

FDP_ACC.1 Subset access control

Hierarchical to: No other components.

FDP_ACC.1.1 The TSF shall enforce the [*discretionary access control SFP*] on [

a) *subject: source domain,*

b) *object: destination domain, and*

c) *operation: release to.*]

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1 The TSF shall enforce the [*discretionary access control SFP*] to objects based on [privilege vectors or firewall mode].

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

1) If there are two or more Dragonfly Guards between the source domain and the destination domain, then

a) [*If the source domain privilege vector has the bit set for the destination domain, then the datagram is released if the MAC check passes, or*

b) *If the destination domain privilege vector has the bit set for the source domain, then the datagram is released if the MAC check passes,*

c) *Else the datagram is not released.*]

2) *If there is only one Dragonfly Guard between the source domain and the destination domain and firewall mode is disabled (i.e., native mode communication is allowed), datagrams are released if they pass the MAC checks.*

3) *If the Dragonfly Guard has Firewall Mode enabled for a port, no datagrams may be received from or released to a Native host in the domain associated with that port].*

Note: A Dragonfly Guard with Firewall Mode enabled for a port will not be able to communicate with hosts attached directly to that port.

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 [*no additional rules]*.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

ITENV.3 Dragonfly Administration System for Setting User Attributes

FDP_ETC.1 Export of user data without security attributes

Hierarchical to: No other components.

FDP_ETC.1.1 The TSF shall enforce the [*mandatory access control SFP*] when exporting user data, controlled under the SFP(s), outside of the TSC.

FDP_ETC.1.2 The TSF shall export the user data without the user data's associated security attributes.

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

ITENV.3 Dragonfly Administration System for Setting User Attributes

Note: FDP_ETC.1 applies only when data is exported to a native host. In this case, the host is in the same security domain and has the same security attributes as the port from which the data is exported.

FDP_IFC.1 Subset information flow control – Mandatory Access Control SFP

Hierarchical to: No other components.

FDP_IFC.1.1 The TSF shall enforce the [*mandatory access control SFP*] on [

a) *Subjects: Dragonfly domains,*

b) *Information: IP datagrams,*

c) *Operation: release from source domain to destination domain*.]

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFF.2 Hierarchical security attributes – Mandatory Access Control SFP

Hierarchical to: FDP_IFF.1

FDP_IFF.2.1 The TSF shall enforce the[*mandatory access control SFP*] based on the following types of subject and information security attributes: [

a) *Security level of the source domain,*

b) *Security level of the destination domain,*

c) *Type of protocol (ARP, RARP, ICMP, UDP, TCP, FTP, and SMTP, and DNS) ,*

d) *Type of request,* response or command,

e) *Writeups enabled,*

f) *ARP Proxy is allowed, and*

g) *RARP Proxy is allowed*.]

FDP_IFF.2.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules, based on the ordering relationships between security attributes hold: [

a) If the security levels of the source domain and destination domain are equal, release the IP datagram.

b) If the security level of the destination domain is greater than the security level of the source domain (writeup), the following rules apply based on the type of protocol:

1) Address Resolution Protocol (ARP)/Reverse Address Resolution Protocol (RARP)

If ARP proxy is allowed, ARP Requests and Responses are allowed.

If the RARP proxy is allowed, RARP Requests and Responses are allowed.

2) If writeups are enabled, the following rules apply:

a) Internet Control Message Protocol (ICMP)

Echo Requests and Time Stamp Requests are allowed.

b) User Datagram Protocol (UDP)

Domain Name Server Requests with the one question flag set are allowed.

c) Transmission Control Protocol (TCP)

Domain Name Server Requests with the one question flag set are allowed.

d) File Transfer Protocol (FTP)

The following FTP commands are allowed: ABOR, ACCT, ALLO, APPE, CWD, MODE, NOOP, PASS, PORT, PWD, QUIT, STOR, STOU, STRU, TYPE, USER, and XPWD.

e) Simple Mail Transfer Protocol (SMTP)

The following SMTP Commands are not allowed: EXPN, HELP, LIST, RETR, STAT, TOP, and TURN. Everything else is allowed.

f) All other messages types are released.

Note: However, since predicted responses are not generated for these message types, any replies to them will be blocked.

c) If the security level of the destination domain is less than the security level of the source domain (writedown), only ARP/RARP requests/responses and predicted messages are released as described below:

When the Dragonfly Unit allows a write up to occur, i.e., releases an ARP/RARP request or an IP datagram to a destination domain at a higher security level, the Dragonfly Guard shall generate a predicted response at the level of the source domain. When the Dragonfly Guard receives an actual response from the destination domain, it shall compare the actual response with the predicted response. If the actual response matches the predicted response, the Dragonfly Unit, shall copy only the fields containing control information (i.e., not user data) specified in the High Level Design from the actual response to the predicted response.

Predicted Responses are listed below by type of protocol. Predicted responses are only released if the actual response matches the predicted response.

1) *Address Resolution Protocol (ARP) /Reverse Address Resolution Protocol (RARP)*

If the ARP proxy is allowed, ARP requests and responses are allowed

If the RARP proxy is allowed, RARP requests and responses are allowed.

2) If writeups are enabled, the following rules apply:

a) *Internet Control Message Protocol (ICMP)*

The following responses are allowed:

ICMP Echo Responses,

ICMP Time Stamp Responses,

ICMP Unreachable Destination,

ICMP Source Quench, and

ICMP Time Exceeded.

b) *User Datagram Protocol (UDP)*

Domain server responses with only one answer are allowed.

c) *Transmission Control Protocol (TCP)*

Domain server responses with only one answer are allowed.

d) *File Transfer Protocol (FTP)*

Predicted responses to the allowed commands that match the actual responses are allowed.

e) *Simple Mail Transfer Protocol (SMTP)*

Predicted responses to the allowed commands that match the actual responses are allowed.]

FDP_IFF.2.3 The TSF shall enforce [*no additional mandatory access control SFP rules*].

FDP_IFF.2.4 The TSF shall provide [*no additional mandatory access control SFP capabilities*].

FDP_IFF.2.5 The TSF shall explicitly authorise an information flow based on the following rules: [*no additional rules].*

FDP_IFF.2.6 The TSF shall explicitly deny an information flow based on the following rules: [*no additional rules].*

FDP_IFF.2.7 The TSF shall enforce the following relationships for any two valid information flow control security attributes:

a) There exists an ordering function that, given two valid security attributes, determines if the security attributes are equal, if one security attribute is greater than the other, or if the security attributes are incomparable; and

b) There exists a “least upper bound” in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is greater than or equal to the two valid security attributes; and

c) There exists a “greatest lower bound” in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is not greater than the two valid security attributes.

Note: The TSF supports the following set of hierarchical security levels: Unclassified, Sensitive But Unclassified (SBU), Confidential, Secret and Top Secret.

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialisation

ITENV.3 Dragonfly Administration System for Setting User Attributes

FDP_ITC.1 Import of user data without security attributes

Hierarchical to: No other components.

FDP_ITC.1.1 The TSF shall enforce the [*mandatory access control SFP*] when importing user data, controlled under the SFP, from outside of the TSC.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [*None]*

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

FMT_MSA.3 Static attribute initialisation

ITENV.3 Dragonfly Administration System for Setting User Attributes

Note: FDP_ITC.1 applies only when data is imported from a native host. In this case, the host is in the same security domain and has the same security attributes as the port on which the data is imported.

FDP_UCT.1 Basic data exchange confidentiality

Hierarchical to: No other components.

FDP_UCT.1.1 The TSF shall enforce the [*mandatory access control SFP*] to be able to [*transmit and receive*] objects in a manner protected from unauthorised disclosure.

Dependencies: [FTP_ITC.1 Inter-TSF trusted channel, or

FTP_TRP.1 Trusted path]

[FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

ITENV.1 Cryptographic Services on the Fortezza Card

Note: Although data confidentiality supports MAC, data confidentiality is provided independently of the mandatory access control SFP.

FDP_UIT.1 Data exchange integrity

Hierarchical to: No other components.

FDP_UIT.1.1 The TSF shall enforce the [*mandatory access control SFP*] to be able to [*transmit and receive*] user data in a manner protected from [*modification, deletion, or insertion*] errors.

FDP_UIT.1.2 The TSF shall be able to determine on receipt of user data, whether [*modification, deletion, or insertion]* has occurred.

Note: Although data integrity supports MAC, data integrity is provided independently of the mandatory access control SFP.

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

[FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]

FDP_UIT.1 Cryptographic Services on the Fortezza Card

5.1.3 Class FIA: Identification and authentication

FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

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

a) User Certificate,

b) Configuration Certificate,

c) Audit Certificate,

d) Certificate Revocation List certificate, and

e) Cryptographic Keys]

Note: The user is the Dragonfly Guard itself. The user attributes contained in the User Certificate, Configuration Certificate, Audit Certificate, and Certificate Revocation List certificate are stored on the User Fortezza Card. These attributes are set by the Dragonfly Administration System. Cryptographic keys are generated by the cryptographic services on the User Fortezza Card during TOE operation.

Dependencies: ITENV.1 Cryptographic Services on Fortezza Card
ITENV.3 Dragonfly Administration System for Setting User Attributes

FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_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
ITENV.1 Cryptographic Services on the Fortezza Card

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.

5.1.4 Class FMT: Security management

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1 The TSF shall restrict the ability to [set] the [audit mask and certificate revocation list] to [the local authority].

Dependencies: FMT_SMR.1 Security roles

ITENV.4 Dragonfly Administration System for Modifying TSF Data

FMT_REV.1 Revocation

Hierarchical to: No other components.

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with [*a Dragonfly Guard*] within the TSC to [*the local authority*].

FMT_REV.1.2 The TSF shall enforce the rules: [*If a certificate appears on a Dragonfly Guard's Certificate Revocation List, the Dragonfly Guard will reject packets originating from a Dragonfly Guard using that Certificate].*

Note: The TSF provides the ability to revoke certificates which contain security attributes.

Dependencies: FMT_SMR.1 Security roles

ITENV.3 Dragonfly Administration System for Setting User Attributes

ITENV.4 Dragonfly Administration System for Modifying TSF Data

FMT_SAE.1 Time-limited authorisation

Hierarchical to: No other components.

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for [*user certificates and cryptographic keys*] to [*the local authority*].

FMT_SAE.1.2 For each of these security attributes, the TSF shall be able to *[not accept packets originating from a Dragonfly Guard using a User Certificate*] after the expiration time for the [*user* certificate or cryptographic key*]* has passed.

Dependencies: FMT_SMR.1 Security roles

FPT_STM.1 Reliable time stamps

ITENV.3 Dragonfly Administration System for Setting User Attributes

FMT_SMR.1 Security roles

Hierarchical to: No other components.

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

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

Note: Certificates for the root authority, root, local authority, and user are stored on the User Fortezza Card for the Dragonfly Guard, but the Dragonfly Guard only assumes the role of User. The TSF associates the Dragonfly Guard user with the User Role when the Dragonfly Guard software logs into the User Fortezza Card using the PIN for the User Certificate.

Dependencies: FIA_UID.1 Timing of identification
ITENV.5 Certificates on the Fortezza Card
ITENV.6 Fortezza Card PINs

5.1.6 Class FPT: Protection of the TOE Security Functions

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.

FPT_ITI.1 Inter-TSF detection of modification

Hierarchical to: No other components.

FPT_ITI.1.1 The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and a remote trusted IT product within the following metric: [*based on the cryptographic services provided by the User Fortezza Card.]*

FPT_ITI.1.2 The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and a remote trusted IT product and [*reject the IP datagram*] if modifications are detected.

Note: IP Datagrams containing TSF Data are either hashed and digitally signed or a checksum is computed and the message and checksum are encrypted using a symmetric key.

Dependencies: ITENV.1 Cryptographic Services on Fortezza Card

FPT_RVM.1 Non-bypassability of the TSP

Hierarchical to: No other components.

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

Dependencies: No dependencies.

FPT_SEP.1 TSF domain separation

Hierarchical to: No other components.

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

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

Note: There is only one security domain on the Dragonfly Guard, the one that the Dragonfly Guard executes its own code in. No other code is executed on a Dragonfly Guard.

Dependencies: No dependencies.

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.

FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to: No other components.

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret [*all security attributes*] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use [*the following rule: the security attributes received from another TOE's TSF (i.e., another Dragonfly Guard) mean the same on the TSF at which it is received*] when interpreting the TSF data from another trusted IT product.

Dependencies: No dependencies.

Note: Dragonfly Guards only interpret TSF data from other Dragonfly Units.

5.1.7 Class FTP: Trusted path/channels

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. .

Note: Dragonfly messages containing TSF Data that needs to be protected from disclosure are encrypted. Dragonfly Messages that require protection from modification but not disclosure such as Association Request and Grant messages are digitally signed, but not encrypted.

FTP_ITC.1.2 The TSF shall permit [either *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 [*communication with another Dragonfly Unit].*

Dependencies: ITENV.1 Cryptographic Services on Fortezza Card

5.1.8 Strength of Function Requirement

The minimum strength of function level for the TOE security functional requirements is SOF-medium.

5.2 TOE SECURITY ASSURANCE REQUIREMENTS

The Security Assurance Requirements for the TOE are the assurance components of Evaluation Assurance Level 2 (EAL2) taken from Part 3 of the Common Criteria. None of the assurance components are refined. The assurance components are listed in Table 5.2.

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

Table 5.2 – EAL2 Assurance Components

5.2.1 Class ACM: Configuration Management

ACM_CAP.2 Configuration items

Objectives

A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using.

Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE.

Dependencies :

No dependencies.

Developer action elements :

ACM_CAP.2.1D The developer shall provide a reference for the TOE.

ACM_CAP.2.2D The developer shall use a CM system.

ACM_CAP.2.3D The developer shall provide CM documentation.

Content and presentation of evidence elements :

ACM_CAP.2.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.2.2C The TOE shall be labelled with its reference.

ACM_CAP.2.3C The CM documentation shall include a configuration list.

ACM_CAP.2.4C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.5C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.2.6C The CM system shall uniquely identify all configuration items.

Evaluator action elements :

ACM_CAP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2 Class ADO: Delivery and Operation

ADO_DEL.1 Delivery procedures

Dependencies :

No dependencies.

Developer action elements :

ADO_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.1.2D The developer shall use the delivery procedures.

Content and presentation of evidence elements :

ADO_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user’s site.

Evaluator action elements :

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

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

Dependencies :

AGD_ADM.1 Administrator guidance

Developer action elements :

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements :

ADO_IGS.1.1C The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.

Evaluator action elements:

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

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.2.3 Class ADV: Development

ADV_FSP.1 Informal functional specification

Dependencies :

ADV_RCR.1 Informal correspondence demonstration

Developer action elements :

ADV_FSP.1.1D The developer shall provide a functional specification.

Content and presentation of evidence elements :

ADV_FSP.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.1.2C The functional specification shall be internally consistent.

ADV_FSP.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV_FSP.1.4C The functional specification shall completely represent the TSF.

Evaluator action elements :

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

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

ADV_HLD.1 Descriptive high-level design

Dependencies:

ADV_FSP.1 Informal functional specification

ADV_RCR.1 Informal correspondence demonstration

Developer action elements :

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

Content and presentation of evidence elements :

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

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

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

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

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

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

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

Evaluator action elements:

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

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

ADV_RCR.1 Informal correspondence demonstration

Dependencies:

No dependencies.

Developer action elements :

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

Content and presentation of evidence elements :

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.

Evaluator action elements :

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

5.2.4 Class AGD: Guidance Documents

AGD_ADM.1 Administrator guidance

Dependencies :

ADV_FSP.1 Informal functional specification

Developer action elements :

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

Content and presentation of evidence elements :

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.

Evaluator action elements:

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

AGD_USR.1 User guidance

Dependencies:

ADV_FSP.1 Informal functional specification

Developer action elements :

AGD_USR.1.1D The developer shall provide user guidance.

Content and presentation of evidence elements :

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.

Evaluator action elements:

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

5.2.5 Class ATE: Tests

ATE_COV.1 Evidence of coverage

Objectives

In this component, the objective is to establish that the TSF has been tested against its functional specification. This is to be achieved through an examination of developer evidence of correspondence.

Application notes

While the testing objective is to cover the TSF, there is no requirement to provide anything to verify this assertion other than an informal mapping of tests to the functional specification and the testing data itself.

Dependencies :

ADV_FSP.1 Informal functional specification

ATE_FUN.1 Functional testing

Developer action elements:

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

Content and presentation of evidence elements :

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

Evaluator action elements:

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

ATE_FUN.1 Functional testing

Objectives

The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation.

Dependencies :

No dependencies.

Developer action elements :

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation of evidence elements :

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.

Evaluator action elements :

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

ATE_IND.2 Independent testing – sample

Objectives

The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes selecting and repeating a sample of the developer tests.

Application notes

The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc.

This component contains a requirement that the evaluator has available test results from the developer to supplement the program of testing. The evaluator will repeat a sample of the developer's tests to gain confidence in the results obtained. Having established such confidence the evaluator will build upon the developer's testing by conducting additional tests that exercise the TOE in a different manner. By using a platform of validated developer test results the evaluator is able to gain confidence that the TOE operates correctly in a wider range of conditions than would be possible purely using the developer's own efforts, given a fixed level of resource. Having gained confidence that the developer has tested the TOE, the evaluator will also have more freedom, where appropriate, to concentrate testing in areas where examination of documentation or specialist knowledge has raised particular concerns.

Dependencies :

ADV_FSP.1 Informal functional specification

AGD_ADM.1 Administrator guidance

AGD_USR.1 User guidance

ATE_FUN.1 Functional testing

Developer action elements :

ATE_IND.2.1D The developer shall provide the TOE for testing.

Content and presentation of evidence elements :

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.

Evaluator action elements :

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.2.6 Class AVA: Vulnerability Assessment

AVA_SOF.1 Strength of TOE security function evaluation

Dependencies :

ADV_FSP.1 Informal functional specification

ADV_HLD.1 Descriptive high-level design

Developer action elements :

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.

Content and presentation of evidence elements :

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.

Evaluator action elements:

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

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

AVA_VLA.1 Developer vulnerability analysis

Objectives

A vulnerability analysis is performed by the developer to ascertain the presence of obvious security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE.

Application notes

The evaluator should consider performing additional tests as a result of potential exploitable vulnerabilities identified during other parts of the evaluation.

Dependencies :

ADV_FSP.1 Informal functional specification

ADV_HLD.1 Descriptive high-level design

AGD_ADM.1 Administrator guidance

AGD_USR.1 User guidance

Developer action elements :

AVA_VLA.1.1D The developer shall perform and document an analysis of the TOE deliverables searching for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2D The developer shall document the disposition of obvious vulnerabilities.

Content and presentation of evidence elements:

AVA_VLA.1.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

Evaluator action elements:

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

AVA_VLA.1.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

5.3 SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT

ITENV.1 Cryptographic Services on Fortezza Card

The Dragonfly Guard relies upon the Fortezza Card to provide the following cryptographic services: secure hash, digital signature, key exchange algorithm, and symmetric key encryption.

ITENV.2 Cryptographic Services Strength of Function (SOF) Requirement

The Dragonfly Guard relies upon the Fortezza Card to meet the Strength of Function (SOF) requirement for the cryptographic services that it provides.

ITENV.3 Dragonfly Administration System for Setting User Attributes

The Dragonfly Guard relies upon the Dragonfly Administration System to configure the system by setting its security attributes and creating the User Fortezza Card. The security attributes of a Dragonfly Guard are set by the local authority on the Dragonfly Administration and the User Certificate on the Dragonfly Guard's User Fortezza Card is signed by the local authority. Note that the Dragonfly Guard installer can check that the Guard's security attributes were set correctly by examining its output at its serial port during initialization.

ITENV.4 Dragonfly Administration System for Modifying TSF Data

The Dragonfly Guard relies upon the Dragonfly Administration System to update Audit Masks and Certificate Revocation Lists. The local authority must update the Audit Mask or Certificate Revocation List on the Dragonfly Administration System and then recreate the User Fortezza Card for one of the Dragonfly Guards that is serving as an Audit Catcher. The Dragonfly Guard receiving the new User Fortezza Guard has to be reinitialized.

ITENV.5 Certificates on the Fortezza Card

The Dragonfly Guard relies on the Fortezza card to store the following certificates: root authority, root, local authority, user, configuration, audit, revocation list, and routing. The first four (root authority, root, local authority, and user) are equivalent to roles. However, the Dragonfly Guard only assumes the user role. The last four are used to store attributes of the user as well as additional non-security policy relevant configuration data for the Guard.

Notes:

The user certificate is signed by the local authority, the local authority certificate is signed by the root, the root certificate is signed by the root authority providing a chain of trust from the user to the root authority.

The local authority role is assumed by the administrator on the Dragonfly Administration System, but this is not part of the Dragonfly Guard TSF.

ITENV.6 Fortezza Card PINs

The Fortezza card requires that the correct PIN for a valid certificate be entered before access is granted to services on the Fortezza card. The Dragonfly Guard software must enter the PIN for the User Certificate.

Note: The administrator must enter the correct PIN for the local authority on the Dragonfly Administration System, but this is not part of the Dragonfly Guard TSF.

6.0 TOE SUMMARY SPECIFICATION

6.1 IT SECURITY FUNCTIONS

6.1.1 Identification and Authentication

IA-1. Dragonfly Guard User Fortezza Card

A User Fortezza Card must be inserted in order for a Dragonfly Guard to start up. The Dragonfly Guard will cease operating if the User Fortezza card is removed. The Fortezza card contains a User Fortezza Certificate that is used to identify the Dragonfly Guard.

IA-2. Fortezza Card Certificate PIN

A user or program must successfully login to a Fortezza Card using the correct PIN for the certificate in order to use Fortezza services. The Dragonfly Guard software enters the Fortezza User Certificate PIN. Dragonfly Guard User Certificates are created on an Administration System by the local authority. The local authority must enter the correct PIN for the local authority certificate in order to login to the Administration System.

IA-3. Source Authentication

Source authentication is performed when one Dragonfly Guard requests an association with another. The source Dragonfly Guard digitally signs the association request and the destination Dragonfly Guard verifies the digital signature.

6.1.2 Associations

ASSOC-1. Association as a Trusted Channel

Dragonfly Guards form an association that provides an Inter-TSF trusted channel between two Dragonfly Guards. No user data is communicated until an association is formed.

ASSOC-2. Digitally Signed Association Request

The originating Dragonfly Guard inserts its User Certificate into an Association Request and digitally signs the association request before releasing it so that other Dragonfly Guards can verify the source of the message.

ASSOC-3. Use of Fortezza Key Exchange Algorithm

When two Dragonfly Guards form an association, they make use of the Fortezza Key Exchange Algorithm to create a symmetric key that is known only to the source and destination Dragonfly Guard.

ASSOC-4. Encryption of User Data

All user data sent between the two Dragonfly Guards is encrypted using the symmetric key generated by the Key Exchange Algorithm.

6.1.3 Discretionary Access Control (DAC)

DAC-1. Privilege Vectors

Dragonfly Guards enforce DAC between the source Dragonfly domain and the destination Dragonfly domain using privilege vectors. Each domain has a privilege vector associated with it. Other Dragonfly Domains are represented by bits in the privilege vector. If either the destination domain bit is set in the source domain's privilege vector, or the source domain bit is set in the destination domain's privilege vector, an association may be formed between hosts in the source domain and the destination domain. DAC checks are performed at the time of association. DAC checks provide the ability to control the release of IP datagrams between Dragonfly Domains at the same security level.

DAC-2. Firewall Mode

If a Dragonfly Domain is not set to Firewall Mode, communication with Native Hosts is allowed. If there is only one Dragonfly Guard between a source host and a destination host and the port is not in Firewall Mode, Privilege Vectors are not checked and IP datagrams are released, assuming that the MAC checks pass.

6.1.4 Security Levels

SL-1. Security Levels

Dragonfly Guards implement the following security levels:

Unclassified,

Sensitive but Unclassified,

Confidential,

Secret,

Top Secret.

SL-2. Dominance Relationships

Top Secret strictly dominates Secret. Secret strictly dominates Confidential. Confidential strictly dominates Sensitive but Unclassified. Sensitive but Unclassified strictly dominates Unclassified.

SL-3. Single Level Ports

Each of the two Ethernet ports on a Dragonfly Guard is configured with a security level for its corresponding Dragonfly Domain. All hosts in the same Dragonfly Domain must be at that same security level and all IP datagrams originating from that domain are labeled at its security level.

6.1.5 Mandatory Access Control (MAC)

MAC-1. Mandatory Access Control Policy.

A Dragonfly Guard will not release IP Datagrams containing User Data from a domain at a higher security level to a domain at a lower security level.

MAC-2. Write Equal

The MAC policy imposes no restrictions on the flow of IP datagrams between Dragonfly Domains at the same level.

MAC-3. FTP Datagrams Supported for Write Up

The following File Transfer Protocol (FTP) commands are allowed, if Write Ups are enabled:

ABOR, APPE, MODE, NOOP, PASS, PORT, PWD, SRTU, STOR, STOU, TYPE, USER

MAC-4. SMTP Datagrams Blocked for Write Up

The following Simple Mail Transfer Protocol (SMTP) commands are always blocked for Write Up, even if Write Ups are enabled:

EXPN, HELP, LIST, RETR, STAT, STT, TOP, TURN

Other SMTP datagrams are allowed if writeups are enabled.

MAC-5. Allowed Information Flows

The Dragonfly Guard can be configured to allow the following control information to be released from a higher security level Dragonfly Domain to a lower security level Dragonfly Domain:

a) ARP and RARP responses

b) ICMP responses

c) UDP and TCP Name Server responses with a single answer, and

d) Anticipated FTP or SMTP messages as described below.

No other information is allowed to flow from a higher level Dragonfly Domain to a lower level Dragonfly Domain.

MAC-6. FTP and SMTP Anticipated Responses

In order for the FTP and SMTP protocols to work, it is necessary for responses to the allowed write up messages to be returned to the originating host. The Dragonfly Guard has implemented a patented write up mechanism of anticipated responses to control the information that can flow from higher level Dragonfly Domains to lower level Dragonfly Domains as responses.

When a Dragonfly Guard releases a message for write up, it creates the anticipated response at the security level of the originating host. When the Dragonfly Guard receives an actual response from the write up message, it compares it to the anticipated response. If the actual response matches the anticipated response, the anticipated response is released to the originating host. If the actual response and anticipated response do not match, nothing is released to the originating host and an audit event message may be generated.

In some cases, it is necessary to copy some fields of control information (such as number of bytes received) from the actual response to the anticipated response. These copied fields allow information to flow from the higher level Dragonfly Domain to the lower level Dragonfly Domain. The TOE documentation identifies the anticipated response for each write up, and the fields and number of bytes that are copied from the actual response to the anticipated response.

MAC-7 ARP/RARP Processing

If the Dragonfly Guard is configured with ARP/RARP processing enabled, the Dragonfly Guard releases ARP and RARP requests and responses for writeup and ARP and RARP requests and anticipated responses for writedown. The Dragonfly Guard can be further configured by defining hosts or ranges of IP addresses for which ARP/RARP processing is explicitly permitted or explicitly denied.

MAC-8 Name Server Requests and Responses

If the Dragonfly Guard is configured with Name Server processing enabled, the Dragonfly Guard releases Name Server requests and responses without performing a mandatory access control (MAC) check. Name Service Requests are allowed from low host to high servers and Responses from high servers to low hosts only if “Write Ups” are enabled; otherwise they are blocked. Name Server Requests from high hosts to low servers are always blocked. (This would be audited as an attempted “Write Down”).

MAC-9. ICMP Requests and Responses

Dragonfly Guards allow the following ICMP requests for writeup:

ICMP Echo Request, and

ICMP Time Stamp Request.

Dragonfly Guards allow the following anticipated ICMP responses for writedown:

ICMP Echo Response,

ICMP Time Stamp Response,

ICMP Unreachable Destination,

ICMP Source Quench, and

ICMP Time Exceeded.

MAC-10. MAC Configuration Options

A Dragonfly Guard can be configured to disallow all flows of control information (except ICMP messages) from a higher security level to a lower security level by doing the following:

a) Disabling write ups,

b) Disabling ARP/RARP Processing, and

c) Disabling Name Server requests and responses.

6.1 6 Data Export and Import

EXP-1. Export of User Data

When User Data is exported to a Native Host, it is exported in unencrypted form without its security level. Data is only exported to Native Hosts within a Dragonfly's domain so the hosts are at the same security level as the port to which the domain is connected.

IMP-1. Import of User Data

When User Data is imported from a Native Host, it is imported with a security level at the level of the Dragonfly Guard's port.

6.1.7 Dragonfly IP Datagrams and Messages

IP-1. Types of IP Datagrams

From the perspective of the protection provided by the Dragonfly Guard, there are four types of IP datagrams:

1) Native IP Datagrams which do not provide integrity or confidentiality protection,
2) Dragonfly Pings,
3) Signed IP Datagrams which provide integrity protection by means of a digital signature, and
4) Encapsulated IP Datagrams which have a checksum for integrity and are encrypted for confidentiality.

IP-2. Native IP Datagrams

Any IP datagram generated by a Native Host is termed a Native IP datagram. The Dragonfly Guard does not provide integrity or confidentiality protection for Native IP datagrams.

IP-3. Dragonfly Ping

Dragonfly pings (i.e., an ICMP request with the last 2 bytes set to the Dragonfly flag (0xdfdf)) are generated by a Guard and not a native host. A Guard may send a Dragonfly TCP SYNC Message first depending upon the initial message from the native host.

IP-4. Signed IP Datagrams

Signed IP datagrams are used to transmit TSF data. Messages are digitally signed by the source Dragonfly Guard and if applicable, the previous intermediate guard. The following types of messages are signed IP datagrams:

1) Association Request
2) Association Grant
3) Association Denial
4) Association Unknown
5) Host Unknown

IP-5. Encapsulated IP Datagrams

For encapsulated datagrams, a checksum is computed and stored in the message. Then the message contents including checksum is encrypted using a symmetric key. All user data is encrypted, before it is transmitted between two Dragonfly Guards. The following types of messages are encapsulated IP datagrams:

1) Type 1 Protected User Datagram (PUD),
2) Type 2 Protected User Datagram (PUD),
3) Audit Event Message
4) Check-in Message
5) Receipt Message
6) Audit Mask Message
7) Certificate Revocation List Message

IP-6. Protected User Datagrams and Security Levels

When User Data is transmitted from one Dragonfly Guard to another Dragonfly Guard, it is transmitted with its associated security level in a Protected User Datagram.

6.1.8 Confidentiality

CONF-1. Confidentiality of User Data

Dragonfly Guards provide confidentiality protection for User Data when it is transmitted between two Dragonfly Guards. User Data is transmitted in a Protected User Datagram (PUD) that is encrypted with a symmetric key known only to the Source and Destination Dragonfly Guards.

6.1.9 Integrity

INT-1. Integrity of User Data

User Data is always transmitted between two Dragonfly Guards in a Protected User Datagram. A checksum is computed and stored in the message, and then the message is encrypted using Cipher Block Chaining Mode (CBC-64) of the Skipjack algorithm.

INT-2. Integrity of TSF Data

TSF Data is transmitted either in digitally signed IP Datagrams or in Encapsulated IP Datagrams for which a checksum is computed and stored in the message before it is encrypted. The digital signature or checksum is checked for integrity by both destination Dragonfly Guards and intermediate Dragonfly Guards. In the case of encrypted checksums, a symmetric key (known only to the source and destination Guards) is used to encrypt the user data and the checksum. If one or more intermediate Guards exists between the source and destination Guards, a second symmetric key (known only to adjacent Guards) is used to encrypt a second checksum so that the intermediate Guards can verify the integrity of the message without decrypting the user data.

6.1.10 Audit

AUDIT-1. Audit Catchers

Any Dragonfly Guard can be specified as an Audit Catcher. An Audit Catcher receives Audit Messages and outputs them through its serial port. The serial port can be connected to a printer, a terminal, or another system to print, display or save the Audit output. The security level of an Audit Catcher's serial port is system high. The "I'm Auditor" field in the Configuration Certificate of the User Fortezza Card determines whether or not a Guard is an Audit Catcher.

AUDIT-2: Audit Required Configuration Option

The local authority specifies whether or not audit will be required when the User Fortezza Card is configured on the Dragonfly Administration System. If audit is required, the Dragonfly Guard will not release any messages if it is unable to form an association with an Audit Catcher.

AUDIT-3. Audit Catcher List

The local authority specifies a list of one to five Audit Catchers required when the User Fortezza Card is configured on the Dragonfly Administration System. The Dragonfly Guard tries the first Audit Catcher on the list and if it does not receive a Receipt Message in the specified time period, it tries the second Audit Catcher on the list. It precedes down the list trying Audit Catchers one at a time until it a Receipt Message is received in the specified time period. If Audit is Required and no Audit Catcher is responding, the Dragonfly Guard stops processing.

AUDIT-4. Audit Catcher Messages

The following messages are either sent to or received from an Audit Catcher:

- Audit Event Message,
- Check-In Message,
- Audit Mask Message,
- Revocation Messages, and
- Receipt Message

Audit Event Messages are sent from Dragonfly Guards to Audit Catchers to report an auditable event.

Check-In Messages are sent from Dragonfly Guards to Audit Catchers upon initialization and periodically thereafter. They contain the Software version, the Audit Mask version, and the Certificate Revocation List Version.

Audit Mask Messages are sent from the Audit Catcher to a Dragonfly Guard to update its Audit Mask.

Revocation Messages are sent from the Audit Catcher to a Dragonfly Guard to update its Certificate Revocation List.

Receipt Messages are sent from the Audit Catcher to a Dragonfly Guard in response to Audit Event Messages and Check-In Messages. Receipt Messages are sent from the Dragonfly Guard to the Audit Catcher in response to Audit Mask Messages and Revocation Messages.

AUDIT-5: Audit Report Fields

The Audit output is in ASCII format. An Audit Report contains the following fields:

- Guard Name: Name of the reporting Dragonfly Unit. Extracted from the Distinguished Name of the User's Fortezza Certificate.
- IP Address: IP Address of the reporting Dragonfly Unit.
- Audit Event Code: A number identifying the type of Audit Event.
- Sender Message Number: A one-up number assigned by the Reporting Dragonfly Unit.
- Date/Time Sent: Year, Month, Day, Hour, Minute, and Second that the Reporting Dragonfly Unit sent the Audit Event Message.
- Date/Time Received: Year, Month, Day, Hour, Minute, and Second that the Audit Catcher received the Audit Report.
- Audit Catcher Message Number: A one-up number assigned by the Audit Catcher upon receipt.

AUDIT-6: Auditable Events

Tables 6.1 lists Audit Event Codes and their corresponding Event Name and Description.

No.	Event Name	Event Description
	Audit Startup	Check-in message from a Guard to its audit catcher; Local status message output by audit catcher to its audit trail.
	Audit Shutdown	Not applicable. Audit is never shutdown once it is started up.
1	Not Used	
2	Closing a Write Up	Reporting Dragonfly Unit detected that a "write up" FTP or SMTP session was closed by the user.
3	Anticipated Message Mismatch	Reporting Dragonfly Unit detected an IP Datagram that was intended for a Write Down, but did not match the anticipated message. The Datagram is not released. Note that this may happen when an unsupported version of FTP or SMTP is encountered.
4	Anticipated Message Not allowed	The user tried to Write Up on a protocol that is not supported, or it may be that the system administrator blocked Write Ups.
5	Anticipated Message Unknown	There was no anticipated message. This represents an attempted Write Down and the transfer is not allowed.
6	Association Request Denied (Reported by Responder)	Reporting Dragonfly Unit has denied another Dragonfly Unit's Association Request. The reason may be that relevant certificates were not yet valid, they were expired, or were revoked. It might also be because the requesting Dragonfly Unit did not have the appropriate privilege (i.e., the DAC check failed.)
7	Association Request Denied (Reported by Initiator)	Reporting Dragonfly Unit has denied another Dragonfly Unit's Association Request. The reason may be that relevant certificates were not yet valid, they were expired, or were revoked. It might also be because the requesting Dragonfly Unit did not have the appropriate privilege (i.e., the DAC check failed.)
8	Association Closed	Reporting Dragonfly Unit detects that an Association has been closed because it has timed out, its Certificate expired or was revoked.

No.	Event Name	Event Description
9	Not used.	
10	Association Granted	Reporting Dragonfly Unit has granted an Association Request.
11	Association Requested	Reporting Dragonfly Unit has requested an Association.
12	Association Unknown	Reporting Dragonfly Unit received a datagram referencing an Association about which the Dragonfly Unit has no information. This normally results from a recycling of Dragonfly Unit and has no security impact.
13	Not used	
14	Not used.	
15	Audit Mask Received	The Audit Mask was received.
16	Not used.	
17	Opening a Write Up Session	Reporting Dragonfly Unit detected that a Write Up FTP or SMTP session was opened for the User.
18	Certificate or Symmetric Key Deleted	Symmetric Keys are routinely deleted when they expire. Certificates are deleted when they are revoked. This is reported by Dragonfly Units when an Association is closed.
19	Not used.	
20	Not used.	
21	Not used.	
22	Not used.	
23	Invalid Signature	Reporting Dragonfly Unit has detected a Dragonfly message (e.g., Association Request, Association Grant, Audit Event Message) that has an invalid digital signature.
24	Not Used	
25	Lost Wait Queue Msg.	A Dragonfly Unit receive an Association Grant or Deny Message and could not find the association request. Relevant only in Intermediate Guards.
26	Not used.	
27	Not Used.	
28	Not used.	
29	Received by non-Audit Catcher	A non-Audit Catcher received a message that should have been sent to an Audit Catcher.
30	Not used.	
31	TPN Registration Complete	Not available in evaluated configuration.
32	Certificate Revocation List Sent	Reporting Dragonfly Unit has sent an updated Certificate Revocation List. The Audit Report identifies the version of the CRL that was sent.
33	Old CRL Version	The Audit Catcher has received a Check In Message referencing an out of date CRL.

No.	Event Name	Event Description
34	Certificate Invalid Start	Reporting Dragonfly Unit has detected a User Fortezza Certificate whose validity period has not yet begun.
35	Certification Expired	Reporting Dragonfly Unit has detected a User Fortezza Certificate whose expiration date/time has passed.
36	Certificate Revoked	Reporting Dragonfly Unit has detected a User Fortezza Certificate that has been revoked.
37	Certificate Invalid	Reporting Dragonfly Unit has detected a User Fortezza Certificate with an invalid digital signature.
38	User Logs onto Companion	Not applicable to Dragonfly Guard
39	User Logs off Companion	Not applicable to Dragonfly Guard
40	Companion changes mode	Not applicable to Dragonfly Guard
41	Audit Catcher Unreachable	Not applicable to Dragonfly Guard
42	Not used.	
43	Security Level Mismatch	Security levels between units are different. This could indicate an error in configuration or a simple error in the Administration System's setup of the deployment.

Table 6.1 – Auditable Events

AUDIT-7. Audit Masks

The Audit Mask is a 256 bit vector with one bit for each auditable event. If an event is to be audited, the bit is turned on in the Audit Mask.

When the local authority configures the User Fortezza Card for a Dragonfly Guard, it can select either Standard, Audit All, or Audit None. Non-standard Audit Masks can be configured by selecting "Edit Audit Masks" on the Administration System.

If the Dragonfly Guard is configured to use the Standard Audit Mask, the audit mask can be updated during normal operations by the Audit Catcher. This means that the selection of auditable events can be changed during normal operations, although it does require inserting an updated User Fortezza Card for the Audit Catcher and re-initializing the Audit Catcher.

AUDIT-8. Audit Mask Management

Audit masks are part of a Dragonfly Guard's initial configuration and are updated by the Audit Catcher. The Audit Mask is identified by name and version number.

The Dragonfly Guard reports the identity of its current Audit Mask to the Audit Catcher in its Check-in Message. The Audit Catcher compares the reported Audit Mask with its current one. If the Dragonfly Guard has an out-of-date Audit Mask, the Audit Catcher sends the current Audit Mask back to the Dragonfly Guard.

Note that Audit Mask messages cannot be sent from an Audit Catcher to a Dragonfly Guard until that Dragonfly Guard checks in with the Audit Catcher and the Audit Mask version is updated. If the check in period is very long, the Guard could miss the auditing of some new events if they occurred while the Audit Catcher was waiting for the Guard to check in. The check in period is stored in the configuration certificate and can be modified by the local authority on the Dragonfly Administration System.

6.1.11 Certificate Revocation

CRL-1. Certificate Revocation List (CRL)

When a Certificate is revoked, the local authority generates a new Certificate Revocation List (CRL) on the Administration System. When the local authority generates a User Fortezza Card on the Dragonfly Administration System, the CRL will be stored in its Certificate Revocation List Certificate. Upon initialization, the Guard uses this CRL unless or until it is updated by the audit catcher.

If the system administrator wishes to update the CRL for a set of guards automatically, this can be done by generating a new User Fortezza Card with the updated CRL for the Guard serving as their Audit Catcher. The new Audit Catcher User Fortezza Card must be generated to add the new CRL, inserted in the Audit Catcher, and the Audit Catcher restarted. When Dragonfly Guards check in with the Audit Catcher, the Audit Catcher sends them the new CRL, if the new CRL is more recent than the Guard's current CRL. Dragonfly Guards will then reject packets originating from Dragonfly Guards using a certificate on the Certificate Revocation List.

CRL-2. CRL Database

Certificates that are revoked are maintained in the Audit Catcher database so that old revoked certificates cannot be used at a later date. Revoked certificates are removed from the CRL only after their certificate expiration date has passed.

6.1.12 Time Stamps

TIME-1. System Time

The system time is taken initially from the User Fortezza Card and then set when the Dragonfly software is loaded.

6.1.13 Security Attributes

ATTR-1. Attribute Definition

Security attributes are set by the local authority on the Administration System and burned into the User Fortezza Card. Security attributes are stored in the User, Configuration, Audit, Certificate Revocation, and Routing Certificates. The contents of these certificates is shown in Tables 6.2 through 6.6.

User Certificate
Certificate Type
Certificate Length
Issuer (i.e., Local Authority) Distinguished Name
Subject (i.e., user’s Distinguished name) Name
Start Time
Expiration Time
Certificate ID
Local Port Security Level
Remote Port Security Level
Local Port Domain ID
Remote Port Domain ID
Local Privilege Vector
Remote Privilege Vector
Public Key
Signature

Table 6.2 – Contents of User Certificate

Audit Certificate
Certificate Type
Certificate Length
Issuer Distinguished Name
Audit Mask ID Number
Expire Time
Audit Mask
Audit Mask Name
Signature

Table 6.3 – Contents of Audit Mask Certificate

Configuration Certificate
Certificate Type
Certificate Length
Issuer Distinguished Name
IP Address – Port A
IP Mask – Port A
IP Gateway – Port A
Do-ARP List – Port A
Don’t-ARP List – Port A
Sec Level – Port A
Firewall – Port A
RARP Avail – Port A
S/W Interrupt – Port A
Network MTU – Port A
Net Type – Port A
[*Same info for Port B as for Port A*] Need to insert this Guard’s name here (as would be registered with a name server … do not confuse this with the Guard’s distinguished name in its User Certificate
IP Address – Audit Catcher 1
Port – Audit Catcher 1
Status – Audit Catcher 1
Hardware Address – Audit Catcher 1
Guard Name – Audit Catcher 1
[*same for Audit Catcher 2*]
[*same for Audit Catcher 3*]
[*same for Audit Catcher 4*]
[*same for Audit Catcher 5*]
Check-In Period
Wait for Receipt Delay
Wait for Association Delay
Association Time to Live
Association Check Period

CGG Version
MSE Port
I'm Auditor
Require Audit Catcher
Receipt Retry
Anticipate Messages
Companion
PPP Install
Time Zone
DS Time
Serial Ports
No Broadcast
Promiscuous
CMP Pass
CMP Default
Key HR2LIV
Default UDP Port
Signature

Table 6.4 – Contents of Configuration Certificate

Certificate Revocation Certificate
Certificate Type
Certificate Length
Issuer Distinguished Name
Revoke List ID Number
Revoked ID Count
Expire Time
Revoked Certificate ID Numbers
Signature

Table 6.5 – Contents of Certificate Revocation Certificate

Routing Certificate
Certificate Type
Certificate Length
Issuer Distinguished Name
Routing ID Number
Number of Entries
IP Address
IP Address Mask
Firewall IP Address
FW Port
Type
Signature

Table 6.6 – Contents of Routing Certificate

ATTR-2. Certificate Expiration

User certificates contain an expiration date. This can be set to any time within one year of the user certificate start date. The default expiration date is one year from the start date.

ATTR-3. Symmetric Key Expiration

There are two expiration times associated with a symmetric key. The first is amount of time allowed for non-use. The second is the total time that the key is valid even when it is being used.

6.1.14 Security Management

SM-1. Types of Certificates

A Dragonfly Fortezza card has the following eight types of certificates:

a) Root Authority (public key),

b) Root signed by Root Authority,

c) Local Authority signed by Root,

d) User signed by Local Authority,

e) Audit signed by Local Authority,

f) Certificate Revocation List signed by Local Authority,

g) Configuration signed by Local Authority, and

h) Routing signed by Local Authority.

The first four: root authority, root, local authority, and user are equivalent to roles. However, the Dragonfly Guard only assumes the User Role.

SM-2. Dragonfly Administration System

The User Fortezza Card for a Dragonfly Guard is configured by the local authority on the Dragonfly Administration System. Although the Dragonfly Administration System has not been included in the evaluated configuration for this evaluation, the local authority can verify that the configuration options of the card are correct, by examining the output at the serial port after the User Fortezza Card is inserted and the Dragonfly Guard is powered up.

SM-3 Management of TSF Data

Initially, a Guard uses the Audit Mask and Certificate Revocation List stored on its own User Fortezza Card. A Dragonfly Guard's Audit Mask and Certificate Revocation List can be updated while it is operating by its Audit Catcher. In order to do this, the local authority must first create a new Fortezza card for the Audit Catcher on the Administration System. When the Audit Catcher is re-initialized and receives a Check-In Message from another Dragonfly Guard, it will send it an Audit Mask Message if the Guard's Audit Mask is out of date or a Revocation Message if the Guard's Certificate Revocation List is out of date.

6.1.15 Inter-TSF Basic Data Consistency

CONS-1. Inter-TSF Data Consistency

Dragonfly Guards inter-operate with other Dragonfly Guards. Most security-relevant values such as security levels and audit masks are constants that are the same on all Dragonfly Guards. The privilege vector is dependent on the configuration of the Dragonfly Domains. Each bit represents a Dragonfly Domain and must be set correctly by the local authority.

6.1.16 System Architecture

SA-1 Non-bypassability of the TSP

The Dragonfly Guard performs access control checks on all messages received from the source domain before releasing them to the destination domain. The source domain and the destination domain are connected to physically separated ports. Messages are processed in the following steps:

1. Incoming message is processed by the TCP/IP stack associated with the source domain.
2. Message is stored in the shared memory area associated with the source domain
3. Message is processed by the security-enforcing code.
4. If a message is to be released, it is stored in the shared memory area associated with the destination domain.
5. Outgoing message is processed by the TCP/IP stack associated with the destination domain.

There is no way for messages to bypass processing by the security-enforcing code.

SA-2. TSF Domain Separation

The Dragonfly Guard maintains a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

- The Dragonfly Guard code is digitally signed and the digital signature is verified before it is installed.
- The only interfaces to the Dragonfly Guard are the PCMCIA card reader, the two Ethernet ports, a serial port, and the power supply.
- No code can be loaded via the network interface.
- No user-developed code runs on the Dragonfly Guard.

6.1.17 Initialization Tests

INIT-1. Self Tests on Initialization

The Dragonfly Guard performs the following initialization procedure when it is powered up:

- The Dragonfly Guard's Central Processing Unit (CPU) runs it own internal Power-On Self Test,
- The Fortezza card runs its own internal self-test,
- The Dragonfly Guard BIOS runs a self-test of the Dragonfly Guard hardware,
- The Dragonfly Guard BIOS checks the checksum and digital signature on the Dragonfly code before loading it from the Ignition Card.
- The Dragonfly code checks the signatures on all the certificates on the Fortezza card before loading the configuration information that they contain.

6.2 ASSURANCE MEASURES

The Dragonfly Guard claims to satisfy the assurance requirements for Evaluation Assurance Level EAL2. The following items will be provided as evaluation evidence to satisfy the EAL2 Assurance Requirements:

a) Configuration Management (CM) Documentation,

b) Functional Specification,

c) High-Level Design,

d) Representation Correspondence,

e) Administrator Guidance,

f) User Guidance,

g) Test Coverage Analysis,

h) Test Documentation, and

i) Vulnerability Analysis

Table 8.11 – Assurance Measures Rationale shows that this evidence is sufficient to meet all of the EAL2 Assurance Requirements.

7.0 PP CLAIMS

The ITT Dragonfly Security Target was not written to address any existing Protection Profile.

8.0 RATIONALE

8.1 SECURITY OBJECTIVES RATIONALE

The first section shows that all of the secure usage assumptions, organizational security policies, and threats to security have been addressed. The second section shows that each IT security objective and each Non-IT security objective counters at least one assumption, policy, or threat. The mappings are straightforward and do not require further explanatory text.

8.1.1 All Assumptions, Policies and Threats Addressed

Table 8.1 shows that all the identified Threats to Security have been addressed. Table 8.2 shows that all of the Organizational Security Policies have been addressed. Table 8.3 shows that all of the Secure Usage Assumptions have been addressed.

No	Threat Name	Threat Description	Objective
1	T.Account	An attempted violation of the TSP may not be traceable to the Guard where it occurred.	O.Accountability
2	T.Acquire_Key	An unauthorized user is able to acquire the key for an encrypted message.	O.Trusted_Channel
3	T.Bypass	A user is able to bypass the security enforcing functions	O.Non-Bypassability
4	T.Card_Lost	A Dragonfly Guard and its associated User Fortezza Card are lost and recovered by a malicious user.	O.Revoke
5	T.Confidential	Data is released in violation of the TSP due to lack of confidentiality during transmission across an unprotected network.	O.Confidentiality
6	T.Excess_Audit	It may not be possible to effectively analyze audit data due to an excessive volume of audit data being recorded.	O.Audit_Select
7	T.Expired	A malicious user is able to use an old User Fortezza Card or an old cryptographic key to gain unauthorized access to information.	O.Expire
8	T.Hardware_Failure	The Dragonfly Guard performs incorrectly due to a hardware failure.	O.Self_Test
9	T.Impersonate	An unauthorized user may attempt to impersonate a Dragonfly Guard.	O.Authen_Source O.Trusted_Channel
10	T.Inconsistent	An incorrect access control decision is made due to a security attributes being interpreted differently on another Dragonfly Guard.	O.Consistency
11	T.Modify_Configuration	The Dragonfly Guard performs incorrectly due to either accidental or intentional modification of configuration data.	O.Verify_Config

No	Threat Name	Threat Description	Objective
12	T.Modify_Data	A message containing User or TSF Data may be modified during transmission.	O.Integrity
13	T.Modify_Software	The Dragonfly Guard performs incorrectly due to either accidental or intentional modification of its software.	O.Self_Test
14	T.No_Need_To_Know	Users have access to data that they have no need to know.	O.DAC
15	T.Quit	A person changes or leaves a job.	O.Revoke
16	T.Sequence	It may not be possible to determine the sequence of security relevant events.	O.Time
17	T.Static_Audit	It may not be possible to record all the security relevant events when suspicious activity is observed due to an inability to dynamically change the set of events that are audited	O.Audit_Select
18	T.Tamper	A malicious user is able to interfere with the execution of the TSF software or modify internal TSF data.	O.Domain_Separation
19	T.Undetected	The occurrence of a suspicious security relevant event may go undetected due to the inability to record security relevant events.	O.Audit
20	T.Write_Down	Information at a higher security level is released on a network at a lower security level.	O.Info_Flow
21	T.Wrong_Level	Exported or imported data may not be properly protected due to the TSF’s inability to correctly associate a security level with data on export or import.	O.Info_Flow O.Single_Level_Port

Table 8.1 – All Threats to Security Addressed by Objectives

Policy Name	Organizational Security Policy	Objective
P.AUDIT	It must be possible to record security relevant actions.	O.Audit
P.DAC	It must be possible to control access between domains at the same security level.	O.DAC
P.MAC	A mandatory access control policy based on hierarchical security levels must be enforced. Information must not be allowed to flow from a higher security level to a lower security level.	O.MAC

Table 8.2 – All Organizational Security Policies Met by Objectives

Assumption Name	Assumption Description	Objective
A. ADMIN	The local authority is trusted to correctly configure User Fortezza Cards.	O-NON-IT.ADMIN
A.ATTACK_LEVEL	Attackers are assumed to have a medium level of expertise, resources, and motivation.	O.SOF
A.CRYPTO_SERVICES	Cryptographic services are provided by the User Fortezza Card.	O.SOF
A.CRYPTO_SOF	The cryptographic algorithms on the Fortezza card are assumed to strong enough to counter at least a medium level of attack.	O.SOF
A.ONLY_PATH	The Guard is assumed to be on the only data path between the two networks connected to its two Ethernet ports.	O-NON-IT.ONLY_PATH
A.PHYSICAL	The Dragonfly Guard is assumed to be protected from physical tampering.	O-NON-IT.PHYSICAL
A.INSTALLER	Authorized installers are assumed to be able to insert the correct User Fortezza Card into the Dragonfly Guard and to connect the correct networks to the local and remote ports.	O.NON-IT.INSTALLER

Table 8.3 – All Secure Usage Assumptions Met by Objectives

8.1.2 All Objectives Necessary

Table 8.4 shows that there are no unnecessary IT security objectives.

No	Objective Name	Objective Description	Threat/Policy/ Assumption
1	O.Accountability	A Guard collecting audit data must associate the security relevant events with the identity of the Guard from which they are reported.	T.Account
2	O.Audit	The Guard must provide an audit capability that can record attempts to bypass the TOE Security Policy.	T.Undetected P.Audit
3	O.Audit_Select	The Guard must be able to change the selection of auditable events during normal operation.	T.Excess_Audit T.Static_Audit
4	O.Authen_Source	A Guard must authenticate itself to another Guard.	T.Impersonate
5	O.Confidentiality	User Data must be protected from disclosure when it is transmitted between two Guards.	T.Confidential
6	O.Consistency	TSF Data must be interpreted consistently by all the Guards within a network.	T.Inconsistent

No	**Objective Name**	**Objective Description**	Threat/Policy/ Assumption
7	O.DAC	The Guard must not release User Data to an unauthorized domain.	T.No_Need_To_Know P.DAC
8	O.Domain_Separation	The Guard must maintain its own domain for execution and ensure that it cannot be interfered with or tampered with by an untrusted subject.	T.Tamper
9	O.Expire	The Guard must provide for the expiration of certificates and keys.	T.Expired
10	O.Info_Flow	The Guard must not release User Data from a higher level domain to a lower level domain.	T.Write_Down T.Wrong_Level P.MAC
11	O.Integrity	User Data and TSF Data must be protected from modification when it is transmitted between two Guards. A Guard must verify the integrity of User Data and TSF data when it is received.	T.Modify_Data
12	O.Non-Bypassability	The Guard must ensure that a packet cannot be released until the security enforcing functions have been invoked and succeed.	T.Bypass
13	O.Revoke	The Guard must provide for the revocation of certificates	T.Card_Lost T.Quit
14	O.Self_Test	The Guard must provide and execute self tests during initial start-up to ensure the its integrity of its hardware and software. .	T.Hardware_Failure T.Modify_Software
15	O.Single_Level_Port	The Guard must assume that all hosts within a Dragonfly Domain are at the same level as the port to which they are connected.	T.Wrong_Level
16	O.SOF	The Guard must be able to meet at least a medium strength of function requirement	A.Attack_Level A.Crypto_Services A.Crypto_SOF
17	O.Time	It must be possible to determine the time of security relevant events.	T.Sequence
18	O.Trusted_Channel	Guards must be able to establish a trusted communication channel between each other.	T.Acquire_Key T.Impersonate
19	O.Verify_Config	A Guard must be able to verify that its configuration certificates have been signed by the local authority.	T.Modify_Configuration

Table 8.4 – All IT Security Objectives Necessary

	Objective Name	**Objective Description**	**Assumption**
20	O-NON-IT.ADMIN	The local authority must be adequately trained on how to configure the User Fortezza Card.	A.ADMIN
21	O-NON-IT.ONLY_PATH	The Dragonfly Guard must be the only data path between the two networks that it is separating.	A.ONLY_PATH
22	O-NON-IT.PHYSICAL	The Dragonfly Guard must be protected from physical tampering.	A.PHYSICAL
23	O-NON-IT.INSTALLER	The Dragonfly Guard Installer must be adequately trained on connecting the Ethernet ports and inserting the correct User Fortezza Card.	A.INSTALLER

Table 8.5 – All Non-IT Security Objectives Necessary

8.2 SECURITY REQUIREMENTS RATIONALE

8.2.1 All Objectives Met by Security Requirements

Table 8.6 shows how the IT security objectives are met. Note that several IT objectives are partially satisfied by the TOE and partially satisfied by the IT environment (i.e., the Dragonfly Administration System and/or the User Fortezza Card.) Since the CC requires that Security Objectives for the TOE be distinguished from Security Objectives for the Environment, the former are prefixed by an “O” and the latter are prefixed by an “O_E”. Security Objectives for the TOE are satisfied by Common Criteria functional components. Security Objectives for the Environment are satisfied by IT requirements for the environment (ITENV.n).

No	Objective Name	Security Requirement
1	O.Accountability	FAU_GEN.1
2	O.Audit	FAU_GEN.1
3	O.Audit_Select	FAU_SEL.1 FMT_MTD.1
3E	O_E.Audit_Select	ITENV.3 ITENV.4
4	O.Authen_Source	FIA_ATD.1 FIA_UID.2 FIA_UAU.2
4E	O_E.Authen_Source	ITENV.1
5	O.Confidentiality	FDP_UCT.1
5E	O_E.Confidentiality	ITENV.1
6	O.Consistency	FPT_TDC.1
7	O.DAC	FDP_ACC.1 FDP_ACF.1 FIA_ATD.1
7E	O_E.DAC	ITENV.3
8	O.Domain_Separation	FPT_SEP.1
9	O.Expire	FMT_SAE.1
9E	O_E.Expire	ITENV.3
10	O.Info_Flow	FDP_IFC.1 FDP_IFF.1 FIA_ATD.1
10E	O_E.Info_Flow	ITENV.3
11	O.Integrity	FDP_UIT.1 FPT_ITI.1
11E	O_E.Integrity	ITENV.1
12	O.Non-Bypassability	FPT_RVM.1
13	O.Revoke	FMT_REV.1 FMT_MTD.1
13E	O_E.Revoke	ITENV.3 ITENV.4
14	O.Self_Test	FPT_AMT.1

No	Objective Name	Security Requirement
15	O.Single_Level_Port	FDP_ETC.1 FDP_ITC.1
15E	O_E.Single_Level_Port	ITENV.3
16	O.SOF	FDP_UIT.1
16E	O_E.SOF	ITENV.1 ITENV.2
17	O.Time	FPT_STM.1
18	O.Trusted_Channel	FTP_ITC.1
18E	O_E.Trusted_Channel	ITENV.1
19	O.Verify_Config	FMT_SMR.1 FPT_AMT.1
19E	O_E.Verify_Config	ITENV.5 ITENV.6

Table 8.6 – Mapping of IT Security Objectives to Functional Requirements

8.2.2 All Functional Components Necessary

No.	Component	Component Name	Objective
1	FAU_GEN.1	Audit data generation	O.Accountability O.Audit
2	FAU_SEL.1	Selective audit	O.Audit_Select
3	FDP_ACC.1	Subset access control	O.DAC
4	FDP_ACF.1	Security attribute based access control	O.DAC
5	FDP_ETC.1	Export of user data without security attributes	O.Single_Level_Port
6	FDP_IFC.1	Subset information flow control	O.Info_Flow
7	FDP_IFF.2	Hierarchical security attributes	O.Info_Flow
8	FDP_ITC.1	Import of user data without security attributes	O.Single_Level_Port
9	FDP_UCT.1	Basic data exchange confidentiality	O.Confidentiality
10	FDP_UIT.1	Data exchange integrity	O.Integrity O.SOF
11	FIA_ATD.1	User attribute definition	O.Authen_Source O.DAC O.Info_Flow
12	FIA_UAU.2	User authentication before any action	O.Authen_Source
13	FIA_UID.2	User identification before any action	O.Authen_Source
14	FMT_MTD.1	Management of TSF Data	O.Audit_Select O.Revoke
15	FMT_REV.1	Revocation	O.Revoke
16	FMT_SAE.1	Time-limited authorisation	O.Expire
17	FMT_SMR.1	Security roles	O.Verify_Config
18	FPT_AMT.1	Abstract Machine Testing	O.Self_Test
19	FPT_ITI.1	Inter-TSF detection of modification	O.Integrity
20	FPT_RVM.1	Non-bypassability of the TSP	O.Non-Bypassability
21	FPT_SEP.1	TSF domain separation	O.Domain_Separation
22	FPT_STM.1	Reliable time stamps	O.Time
23	FPT_TDC.1	Inter-TSF basic TSF data consistency	O.Consistency
24	FTP_ITC.1	Inter-TSF Trusted Channel	O.Trusted_Channel

Table 8.7 – Mapping of Functional Requirements to IT Security Objectives

No.	Requirement	Requirement for the IT Environment	Objective
1	ITENV.1	Cryptographic Services on Fortezza Card	O_E.Authen_Source O_E.Confidentiality O_E.Integrity O_E.Trusted_Channel O_E.SOF
2	ITENV.2	Cryptographic Services Strength of Function (SOF) Requirement	O_E.SOF
3	ITENV.3	Dragonfly Administration System for Setting User Attributes	O_E.Audit_Select O_E.DAC O_E.Expire O_E.Info_Flow O_E.Revoke O_E.Single_Level_Port
4	ITENV.4	Dragonfly Administration System for Modifying TSF Data	O_E.Audit_Select O_E.Revoke
5	ITENV.5	Certificates on the Fortezza Card	O_E.Verify_Config
6	ITENV.6	Fortezza Card PINs	O_E.Verify_Config

Table 8.8 – Mapping of IT Environment Requirements to IT Security Objectives

8.2.3 Satisfaction of Dependencies

Table 8.8 shows the dependencies between the functional requirements. In two cases, the dependency is satisfied by a component that is hierarchical to the required component: FDP_IFF.1 satisfied by FDP_IFF.2, and FIA_UID.1 satisfied by FIA_UID.2. This is indicated in the table by an "(H)" following the reference line number. All of the dependencies are satisfied except FMT_MSA.3. This functionality is provided by the Dragonfly Administration System. The FMT_MSA.3 functionality is provided by two requirements that are satisfied by the IT Environment: ITENV.3: Dragonfly Administration System for Setting User Attributes and ITENV.4 Dragonfly Administration System for Modifying TSF Data, These dependencies have been added to Table 8.8. Also, see the next section on Use of the Dragonfly Administration System.

No.	Component	Component Name	Dependencies	Reference
1	FAU_GEN.1	Audit data generation	FPT_STM.1	22
2	FAU_SEL.1	Selective audit	FAU_GEN.1 FMT_MTD.1 ITENV.3 ITENV.4	1 14
3	FDP_ACC.1	Subset access control	FDP_ACF.1	4
4	FDP_ACF.1	Security attribute based access control	FDP_ACC.1 FMT_MSA.3 ITENV.3	3 none

No.	Component	Component Name	Dependencies	Reference
5	FDP_ETC.1	Export of user data without security attributes	[FDP_ACC.1 or FDP_IFC.1] ITENV.3	3 6
6	FDP_IFC.1	Subset information flow control	FDP_IFF.1	7 (H)
7	FDP_IFF.2	Hierarchical security attributes	FDP_IFC.1 FMT_MSA.3 ITENV.3	6 none
8	FDP_ITC.1	Import of user data without security attributes	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.3 ITENV.3	3 6 none
9	FDP_UCT.1	Basic data exchange confidentiality	[FTP_ITC.1 or FTP_TRP.1] [FDP_ACC.1 or FDP_IFC.1] ITENV.1	24 - 3 6
10	FDP_UIT.1	Data exchange integrity	[FDP_ACC.1 or FDP_IFC.1] FTP_ITC.1 ITENV.1	3 6 24
11	FIA_ATD.1	User attribute definition	None ITENV1 ITENV.3	-
12	FIA_UAU.2	User authentication before any action	None ITENV.1	-
13	FIA_UID.2	User identification before any action	None	-
14	FMT_MTD.1	Management of TSF Data	FMT_SMR.1 ITENV.4	17
15	FMT_REV.1	Revocation	FMT_SMR.1 ITENV.3 ITENV.4	17
16	FMT_SAE.1	Time-limited authorisation	FMT_SMR.1 FPT_STM.1 ITENV.3	17 22
17	FMT_SMR.1	Security roles	FIA_UID.1 ITENV.5 ITENV.6	13(H)
18	FPT_AMT.1	Abstract Machine Testing	None	-
19	FPT_ITI.1	Inter-TSF detection of modification	None ITENV.1	-
20	FPT_RVM.1	Non-bypassability of the TSP	None	-
21	FPT_SEP.1	TSF domain separation	None	-
22	FPT_STM.1	Reliable time stamps	None	-
23	FPT_TDC.1	Inter-TSF basic TSF data consistency	None	-
24	FTP_ITC.1	Inter-TSF Trusted Channel	None ITENV.1	-

Table 8.9 – Functional Requirements Dependencies

8.2.4 Use of the Dragonfly Administration System

The Dragonfly Administration System is outside of the evaluated configuration for the Dragonfly Guard. However, the Dragonfly Administration is used to create the User Fortezza Card for the Dragonfly Guard. The User Fortezza Card contains four certificates: User Certificate, Configuration Certificate, Audit Certificate, and Certificate Revocation List which contain the security attributes for the Dragonfly Guard. It was deemed acceptable for the Dragonfly Administration to be outside of the evaluated configuration, even though the Guard depends on it to set its security attributes, because the Dragonfly Guard installer can check that the values on the User Fortezza Card were set correctly by examining the output at its serial port during initialization.

Because of the way the Dragonfly Guard operates, the Audit Mask and the Certificate Revocation List are both user attributes and TSF data. When a guard is first initialized, it uses the audit mask and certificate revocation list on its own user Fortezza card. However, when there are multiple Dragonfly Guards in a Dragonfly deployment, they periodically exchange audit masks and certificate revocation lists, and each Guard updates itself with the most current values which may come from another Dragonfly Guard. When one Dragonfly Guard updates the Audit Mask or Certificate Revocation List of another Guard, they are considered TSF data.

Two requirements to be satisfied by the IT Environment: ITENV.3: Dragonfly Administration System for Setting User Attributes and ITENV.4 Dragonfly Administration System for Modifying TSF Data have been included in the Security Target to address the dependencies of the Dragonfly Guard on the Dragonfly Administration System. Also, the requirements ITENV.3 and ITENV.4 are used instead of FMT_MSA.3, because the functionality for this requirement is provided by the environment (i.e., the Dragonfly Administration System) rather than the TSF.

8.2.5 Auditable Events Rationale

The auditable events provided by the Dragonfly Guard were reviewed against the auditable events for the minimal or basic level of audit for the functional requirements. It was found that the Dragonfly Guard provided auditable events for the applicable functionality in all areas except for export, import, confidentiality, and integrity. It was decided that it would not be appropriate for the Guard to audit these activities, since all User Data messages sent between two guards have an integrity check applied, are encrypted for confidentiality, and are imported and exported from both guards. These are routine events for the Dragonfly Guard and not appropriate for auditing. Therefore, “not specified” was selected for the level of audit, and all the auditable events were listed.

8.2.6 Mutual Support Rationale

Mutual support is provided by having requirements that meet the dependencies requirements of other requirements. In addition, functional requirements for FPT_RVM.1, Non-bypassability and FPT_SEP.1, Domain Separation have been provided to prevent bypassing or interference with the implementation of the other functional requirements.

8.2.7 Strength of Function Rationale

A Strength of Function level of SOF-Medium counters the assumed attack level of medium. The strength of function requirement is met by using the cryptographic services provided by the User Fortezza Card.

8.2.8 Assurance Requirements Rationale

The Dragonfly Guard claims to satisfy the requirements for EAL2 and no additional assurance requirements. Although the Dragonfly Guard is designed to meet the assurance requirements of a higher assurance level, the highest priority for now is to have it complete an independent evaluation as quickly as possible. Assuming there is an interim period between when the Dragonfly Guard completes its EAL2 evaluation and when it completes its evaluation for a higher assurance level, procedural controls will be used to reduce risk during this period.

The assurance requirements for EAL2 have been specified to be mutually supportive and internally consistent.

8.3 TOE SUMMARY SPECIFICATION RATIONALE

8.3.1 All TOE Security Functional Requirements Satisfied

Table 8.9 shows that the IT Security Functions in the TOE Summary Specification (TSS) address all of the TOE Security Functional Requirements.

Functional Component	**Functional Requirement**	**TSS Ref.**	**IT Security Function**
FAU_GEN.1	Audit data generation	AUDIT-1	Audit Catchers
		AUDIT-2	Audit Required Configuration Option
		AUDIT-3	Audit Catcher List
		AUDIT-4	Audit Catcher Messages
		AUDIT-5	Audit Report Fields
		AUDIT-6	Auditable Events
FAU_SEL.1	Selective audit	AUDIT-6	Auditable Events
		AUDIT-7	Audit Masks
		AUDIT-8	Audit Mask Management
		SM-3	Management of TSF Data
FDP_ACC.1	Subset access control	DAC-1	Privilege Vectors
		DAC-2	Firewall Mode
FDP_ACF.1	Security attribute based access control	DAC-1	Privilege Vectors
		DAC-2	Firewall Mode
		IP-2	Native Datagrams
FDP_ETC.1	Export of user data without security attributes	EXP-1	Export of User Data
		SL-3	Single Level Ports

Functional Component	**Functional Requirement**	**TSS Ref.**	**IT Security Function**
FDP_IFC.1	Subset information flow control	MAC-1	Mandatory Access Control Policy
		MAC-2	Write Equal
		MAC-3	FTP Datagrams Supported for Write Up
		MAC-4	SMTP Datagrams Blocked for Write Up
		MAC-5	Allowed Information Flows
		MAC-6	FTP and SMTP Anticipated Responses
		MAC-7	ARP/RARP Requests and Responses
		MAC-8	Name Server Requests and Responses
		MAC-9	ICMP Requests and Responses
		MAC-10	MAC Configuration Options
FDP_IFF.2	Hierarchical security attributes	SL-1	Security Levels
		SL-2	Dominance Relationships
		SL-3	Single Level Ports
		MAC-1	Mandatory Access Control Policy
		MAC-2	Write Equal
		MAC-3	FTP Datagrams Supported for Write Up
		MAC-4	SMTP Datagrams Blocked for Write Up
		MAC-5	Allowed Information Flows
		MAC-6	FTP and SMTP Anticipated Responses
		MAC-7	ARP/RARP Requests and Responses
		MAC-8	Name Server Requests and Responses
		MAC-9	ICMP Requests and Responses
		MAC-10	MAC Configuration Options
		IP-6	Protected User Datagrams and Security Levels

Functional Component	Functional Requirement	TSS Ref.	IT Security Function
FDP_ITC.1	Import of user data without security attributes	IMP-1	Import of User Data
		SL-3	Single Level Ports
FDP_UCT.1	Basic data exchange confidentiality	ASSOC-3	Use of Fortezza Key Exchange Algorithm
		ASSOC-4	Encryption of User Data
		IP-1	Types of IP Datagrams
		IP-5	Encapsulated Datagrams
		CONF-1	Confidentiality of User Data
FDP_UIT.1	Data exchange integrity	IP-1	Types of IP Datagrams
		IP-5	Encapsulated Datagrams
		INT-1	Integrity of User Data
FIA_ATD.1	User attribute definition	ATTR-1	Attribute Definition
		SM-2	Dragonfly Administration System
FIA_UAU.2	User authentication before any action	ASSOC-2	Digitally Signed Association Request
		IA-1	Dragonfly Guard User Fortezza Card
		IA-2	Fortezza Card Certificate PIN
		IA-3	Source Authentication
FIA_UID.2	User identification before any action	IA-1	Dragonfly Guard User Fortezza Card
		IA-2	Fortezza Card Certificate PIN
		IA-3	Source Authentication
FMT_MTD.1	Management of TSF data	SM-3	Management of TSF data
FMT_REV.1	Revocation	CRL-1	Certificate Revocation List (CRL)
		CRL-2	CRL Database
		SM-3	Management of TSF Data
FMT_SAE.1	Time-limited authorisation	ATTR-2	Certificate Expiration
		ATTR-3	Symmetric Key Expiration
FMT_SMR.1	Security roles	IA-1	Dragonfly User Fortezza Card
		IA-2	Fortezza Card Certificate PIN
		SM-1	Types of Certificates
FPT_AMT.1	Abstract Machine Testing	INIT-1	Self Tests on Initialization
FPT_ITI.1	Inter-TSF detection of modification	ASSOC-2	Digitally Signed Association Request
		ASSOC-3	Use of Fortezza Key Exchange Algorithm
		IP-1	Types of IP Datagrams
		IP-4	Signed Datagrams
		IP-5	Encapsulated Datagrams
		INT-2	Integrity of TSF Data

Functional Component	**Functional Requirement**	**TSS Ref.**	**IT Security Function**
FPT_RVM.1	Non-bypassability of the TSP	SA-2	Non-bypassability of the TSP
FPT_SEP.1	TSF domain separation	SA-2	TSF Domain Separation
FPT_STM.1	Reliable time stamps	TIME-1	System Time
FPT_TDC.1	Inter-TSF basic TSF data consistency	CONS-1	Inter-TSF data Consistency
FTP_ITC.1	Inter-TSF Trusted Channel	ASSOC-1	Association as a Trusted Channel
		ASSOC-2	Digitally Signed Association Request
		ASSOC-3	Use of Fortezza Key Exchange Algorithm
		ASSOC-4	Encryption of User Data

Table 8.10 – Mapping of Functional Requirements to TOE Summary Specification

8.3.2 All TOE Summary Specification (TSS) Functions Necessary

Table 8.10 shows that all of the IT Security Functions in the TOE Summary Specification (TSS) help meet TOE Security Functional Requirements.

TSS **Ref No**	**IT Security Function**	**Functional Component**	**Functional Requirement**
IA-1	Dragonfly Guard User Fortezza Card	FIA_UID.2	User identification before any action
		FIA_UAU.2	User authentication before any action
		FMT_SMR.1	Security Roles
IA-2	Fortezza Card Certificate PIN	FIA_UID.2	User identification before any action
		FIA_UAU.2	User authentication before any action
		FMT_SMR.1	Security Roles
IA-3	Source Authentication	FIA_UID.2	User identification before any action
		FIA_UAU.2	User authentication before any action
ASSOC-1	Association as a Trusted Channel	FTP_ITC.1	Inter-TSF trusted channel
ASSOC-2	Digitally Signed Association Request	FIA_UAU.2	User authentication before any action
		FPT_ITI.1	Inter-TSF detection of modification
		FTP_ITC.1	Inter-TSF trusted channel
ASSOC-3	Use of Fortezza Key Exchange Algorithm	FDP_UCT.1	Basic data exchange confidentiality
		FPT_ITI.1	Inter-TSF detection of modification
		FTP_ITC.1	Inter-TSF trusted channel
ASSOC-4	Encryption of User Data	FDP_UCT.1	Basic data exchange confidentiality
		FTP_ITC.1	Inter-TSF trusted channel

TSS **Ref No**	**IT Security Function**	**Functional Component**	**Functional Requirement**
DAC-1	Privilege Vectors	FDP_ACC.1	Subset access control
		FDP_ACF.1	Security attribute based access control
DAC-2	Firewall Mode	FDP_ACC.1	Subset access control
		FDP_ACF.1	Security attribute based access control
SL-1	Security Levels	FDP_IFF.2	Hierarchical security attributes
SL-2	Dominance Relationships	FDP_IFF.2	Hierarchical security attributes
SL-3	Single Level Ports	FDP_IFF.2	Hierarchical security attributes
		FDP_ETC.1	Export of user data without security attributes
		FDP_ITC.1	Import of user data without security attributes
MAC-1	Mandatory Access Control Policy	FDP_IFC.1	Subset information flow control
		FDP_IFF.2	Hierarchical security attributes
MAC-2	Write Equal	FDP_IFC.1	Subset information flow control
		FDP_IFF.2	Hierarchical security attributes
MAC-3	FTP Datagrams Supported for Write Up	FDP_IFC.1	Subset information flow control
		FDP_IFF.2	Hierarchical security attributes
MAC-4	SMTP Datagrams Blocked for Write Up	FDP_IFC.1	Subset information flow control
		FDP_IFF.2	Hierarchical security attributes
MAC-5	Allowed Information Flows	FDP_IFC.1	Subset information flow control
		FDP_IFF.2	Hierarchical security attributes
MAC-6	FTP and SMTP Anticipated Responses	FDP_IFC.1	Subset information flow control
		FDP_IFF.2	Hierarchical security attributes
MAC-7	ARP/RARP Requests and Responses	FDP_IFC.1	Subset information flow control
		FDP_IFF.2	Hierarchical security attributes
MAC-8	Name Server Requests and Responses	FDP_IFC.1	Subset information flow control
		FDP_IFF.2	Hierarchical security attributes
MAC-9	ICMP Requests and Responses	FDP_IFC.1	Subset information flow control
		FDP_IFF.2	Hierarchical security attributes

TSS **Ref No**	**IT Security Function**	**Functional Component**	**Functional Requirement**
MAC-10	MAC Configuration Options	FDP_IFC.1	Subset information flow control
		FDP_IFF.2	Hierarchical security attributes
EXP-1	Export of User Data	FDP_ETC.1	Export of user data without security attributes
IMP-1	Import of User Data	FDP_ITC.1	Import of user data without security attributes
IP-1	Types of IP Datagrams	FDP_UCT.1	Basic data exchange confidentiality
		FDP_UIT.1	Data exchange integrity
		FPT_ITI.1	Inter-TSF detection of modification
IP-2	Native Datagrams	FDP_ACF.1	Security attribute based access control
IP-3	Dragonfly Pings	FDP_IFF.2	Hierarchical security attributes
IP-4	Signed Datagrams	FPT_ITI.1	Inter-TSF detection of modification
IP-5	Encapsulated Datagrams	FDP_UCT.1	Basic data exchange confidentiality
		FDP_UIT.1	Data exchange integrity
		FPT_ITI.1	Inter-TSF detection of modification
IP-6	Protected User Datagrams and Security Levels	FDP_IFF.2	Hierarchical security attributes
CONF-1	Confidentiality of User Data	FDP_UCT.1	Basic data exchange confidentiality
INT-1	Integrity of User Data	FDP_UIT.1	Data exchange integrity
INT-2	Integrity of TSF Data	FPT_ITI.1	Inter-TSF detection of modification
AUDIT-1	Audit Catchers	FAU_GEN.1	Audit Data Generation
AUDIT-2	Audit Required Configuration Option	FAU_GEN.1	Audit Data Generation
AUDIT-3	Audit Catcher List	FAU_GEN.1	Audit Data Generation
AUDIT-4	Audit Catcher Messages	FAU_GEN.1	Audit Data Generation
AUDIT-5	Audit Report Fields	FAU_GEN.1	Audit Data Generation
AUDIT-6	Auditable Events	FAU_GEN.1	Audit Data Generation
		FAU_SEL.1	Selective Audit
AUDIT-7	Audit Masks	FAU_SEL.1	Selective Audit
AUDIT-8	Audit Mask Management	FAU_SEL.1	Selective Audit

TSS **Ref No**	**IT Security Function**	**Functional Component**	**Functional Requirement**
CRL-1	Certificate Revocation List (CRL)	FMT_REV.1	Revocation
CRL-2	CRL Database	FMT_REV.1	Revocation
TIME-1	System Time	FPT_STM.1	Reliable time stamps
ATTR-1	Attribute Definition	FIA_ATD.1	User attribute definition
ATTR-2	Certificate Expiration	FMT_SAE.1	Time-limited authorisation
ATTR-3	Symmetric Key Expiration	FMT_SAE.1	Time-limited authorisation
SM-1	Types of Certificates	FMT_SMR.1	Security Roles
SM-2	Dragonfly Administration System	FIA_ATD.1	User attribute definition
SM-3	Management of TSF Data	FAU_SEL.1	Selective Audit
		FMT_MTD.1	Management of TSF Data
		FMT_REV.1	Revocation
CONS-1	Inter-TSF Data Consistency	FPT_TDC.1	Inter-TSF basic TSF data consistency
SA-1	Non-bypassability of the TSP	FPT_RVM.1	Non-bypassability of the TSP
SA-2	TSF Domain Separation	FPT_SEP.1	TSF Domain Separation
INIT-1	Self Tests on Initialization	FPT_AMT.1	Abstract Machine Testing

Table 8.11 – Mapping of TOE Summary Specification to Functional Requirements

8.3.3 Assurance Measures Rationale

Component	Component Title	Evidence Requirements	How Sati
ACM_CAP.2	Configuration items	CM Documentation	Guard TC Managem
ADO_DEL.1	Delivery procedures	Delivery Procedures	Dragonfly Manual
ADO_IGS.1	Installation, generation, and start-up procedures	Installation, generation, and start-up procedures	Dragonfly Manual
ADV_FSP.1	Informal functional specification	Functional Specification	Dragonfly Functiona
ADV_HLD.1	Descriptive high-level design	High-Level Design	Dragonfly Level Des
ADV_RCR.1	Informal correspondence demonstration	Representation Correspondence	Dragonfly Correspor Demonstr
AGD_ADM.1	Administrator guidance	Administrator Guidance	Dragonfly Manual
AGD_USR.1	User guidance	User Guidance	Dragonfly Manual
ATE_COV.1	Evidence of coverage	Test Coverage Analysis	Dragonfly Correspor Demonstr
ATE_FUN.1	Functional testing	Test Documentation	Test Plans Test Resu
ATE_IND.2	Independent testing – sample	TOE for Testing	TOE for T
AVA_SOF.1	Strength of TOE security function evaluation	Not applicable	Vulnerabil Dragonfly
AVA_VLA.1	Developer vulnerability analysis	Vulnerability Analysis	Vulnerabil Dragonfly

Table 8.12– Assurance Measures Rationale

8.4 PP CLAIMS RATIONALE

Not applicable.

APPENDIX A ACRONYMS

ARP	Address Resolution Protocol
CBC	Cipher-Block Chaining
CC	Common Criteria for IT Security Evaluation
CM	Configuration Management
CPU	Central Processing Unit
CRL	Certificate Revocation List
DAC	Discretionary Access Control
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
FTP	File Transfer Protocol
ICMP	Internet Control Message Protocol
ID	Identification
INE	In-line Encryption
IP	Internet Protocol
IT	Information Technology
IWG	Internet Gateway
KEA	Key Exchange Algorithm
LAN	Local Area Network
MAC	Mandatory Access Control
MLS	Multi-Level Secure
NSA	National Security Agency
PC	Personal Computer
PCMCIA	Personal Computer Memory Card International Association
PIN	Personal Identification Number
PP	Protection Profile
PUD	Protected User Datagram
RARP	Reverse Address Resolution Protocol

SBU	Sensitive But Unclassified
SF	Security Function
SFP	Security Function Policy
SHA	Secure Hash Algorithm
SMTP	Simple Mail Transfer Protocol
ST	Security Target
TCP	Transport Control Protocol
TNS	Tactical Name Server
TOE	Target of Evaluation
TPN	Tactical Packet Network
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy
UDP	User Datagram Protocol

APPENDIX B REFERENCES

Dragonfly Documents

DF_AUM ITT Industries, *The Dragonfly Administration User Manual*, Release 1 (v2.02), June 1998.

DF_CM ITT Industries, *Guard TOE Configuration Management, Reference 98-016c, 27 August 1998.*

DF_CORR ITT Industries, *Dragonfly Guard Informal Correspondence Demonstration, , Version 1.3, 27 August 1998.*

DF_FUNC ITT Industries, *Dragonfly Guard Informal Functional Specification, Version 1.0, 29 July 1998.*

DF_GUM ITT Industries, *The Dragonfly Guard User Manual*, Release 1 (v2.01), June 1998.

DF_HLD ITT Industries, *Dragonfly Descriptive High Level Design Document, Version 1.2, 4 September 1998.*

DF_POP Arca, *ITT Dragonfly Philosophy of Protection*, ATR-97003, 15 April 1997.

DF_SCONOPS ITT Industries, *Security Concept of Operations for the ITT Dragonfly*, DRAFT

DF_Test ITT Industries, *Dragonfly Test Plan/Procedures, Version 2.5, 4 September 1998*

DF_THOPS ITT Industries, *Dragonfly Theory of Operations*, Version 2.0, DRAFT, 3 June 1998.

DF_TO ITT Industries, *Dragonfly Technical Overview*, Document DF174, Issue 1.2, 19 March 1998.

DF_VA ITT Industries, *Vulnerability Analysis of the Dragonfly Guard, Version 2.4, 22 July 1998.*

Standards

CCITSE *Common Criteria for Information Technology Security Evaluation*, CCIB-98-026, Version 2.0, May 1998.

ST_Guide Donaldson, Murray G., *Guide for the Production of PPs and STs*, Version 0.6, 8 July 1998, ISO/IEC JTC 1/SC 27/WG 3 N452.

U. S. Government Documents

TFW_PP *US government Traffic Filter Firewall Protection Profile for Low Risk Environments*, Version 1.0, December 1997.

Fortezza National Security Agency, Workstation Security Products, *Fortezza Application Implementors Guide,* Revision 1.52, 5 March 1996.