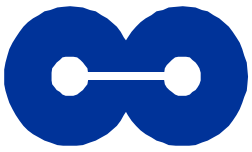
# ITT INDUSTRIES
# DRAGONFLY COMPANION
# SECURITY TARGET
# VERSION 1.5

---

Kristina Rogers
Savith Kandala

29 October 1999



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

# 1.0 SECURITY TARGET INTRODUCTION

## 1.1 SECURITY TARGET IDENTIFICATION

TOE Identification:

ITT Industries Dragonfly Companion, Version 3.02, Build 129

Windows 95 Operating System, Version 4.00.950

ITT Industries Dragonfly Guard Model G1.2 running Dragonfly software release 3.0, Build 980908.1509

ST Identification: ITT Industries Dragonfly Companion Security Target, Version 1.5

Assurance level: EAL2

Registration: <To be filled in upon registration>

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

## 1.2 SECURITY TARGET OVERVIEW

The ITT Dragonfly Companion is a network security software product that is used primarily to protect the host it resides on from the unsecured network to which the host is connected. The ITT Dragonfly Companion is installed on a Personal Computer (PC) with the Windows 95 operating system. The Companion can inter-operate with other ITT Dragonfly Companions and ITT Dragonfly Guards. Dragonfly Guards and Dragonfly Companions are collectively referred to as Dragonfly Units. The Companion offers a configuration option that allows it to operate with native hosts (i.e., hosts that are not protected by Dragonfly Units.)

The Dragonfly Guard is a network security device that provides the same security services as a Dragonfly Companion. Since the Guard and Companion are similar, the Security Target (ST) of the Companion is similar to the ST of the Guard and the ST of the Guard is described in Dragonfly Guard Security Target [DF_GST].

The Dragonfly Companion operates on standard Internet Protocol (IP) datagrams. The Dragonfly Companion provides the following security services: mandatory access control, discretionary access control, confidentiality, integrity, source authentication, and audit. A Dragonfly Companion cannot be designated as an Audit Catcher, but can store its audit messages on the PC it protects and send audit reports to a Dragonfly Guard that is acting as an Audit Catcher.

## 1.3 COMMON CRITERIA CONFORMANCE

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

# 2.0 TOE DESCRIPTION

## 2.1 EVALUATION SCOPE

The Target of Evaluation (TOE) consists of:

- ITT Dragonfly Companion, Version 3.02, Build 129,
- Windows 95 Operating System, version 4.00.950, and
- ITT Industries Dragonfly Guard Model G1.2 running Dragonfly software release 3.0, Build 980908.1509.

The ITT Dragonfly Companion was evaluated on the following hardware configuration:

- Pentium 75 MHz processor
- 8 MB Random Access Memory
- 1 502 MB hard disk drive
- 1 Floppy Disk Drive
- 1 LinkSys Combo PCMCIA Ethernet Card

No untrusted users are allowed on the Windows 95 operating system and no other programs may be installed on the ITT Dragonfly Companion host PC. These first two assumptions are enforced procedurally and are addressed in the Administrator Guidance. The Windows 95 operating system also must be configured so that it accepts only Internet Protocol (IP) datagrams, It is assumed that Windows 95 cannot be attacked through network protocols below the IP layer. The two assumptions that Windows 95 can be configured to only accept IP datagrams and that Windows 95 cannot be attacked through network protocols below the IP layer will be verified as part of the evaluation. The ITT Dragonfly Companion makes no other assumptions about the security functionality provided by Windows 95,

The ITT Dragonfly Guard configured to serve an Audit Catcher is part of the TOE. The ITT Dragonfly Guard has completed its EAL2 evaluation. The Companion relies on the Guard configured as an audit catcher to receive its audit records. In addition, the Guard sends messages to the Companion to update its Certificate Revocation List and Audit Mask. The security functionality provided by the Guard is documented in the ITT Industries Dragonfly Guard Security Target [DF_GST].

Although they are not part of the Target of Evaluation, the ITT Dragonfly Companion relies upon the following systems as part of the Information Technology (IT) environment:

- User Fortezza Card and the
- ITT Dragonfly Administration System.

The correct operation of the ITT Dragonfly Administration System can be verified by checking its output upon initialization to verify that the User Fortezza Card has been properly configured.

## 2.2 PRODUCT DESCRIPTION

The ITT Dragonfly Companion is a network security software product that uses National Security Agency (NSA) Fortezza Cards to provide multi-level secure (MLS) services to Internet Protocol (IP) networks. Dragonfly Companions allow users to send classified or commercial proprietary information over any IP

based network without worrying about that information being available to anyone other than the intended recipient. Dragonfly Companions operate on standard IP datagrams.

Dragonfly Companion software is installed on a host Personal Computer (PC) that has Windows 95 as its operating system. The host PC must have a PCMCIA slot and a PCMCIA card reader. Dragonfly Companions require a PCMCIA User Fortezza Card to operate. The User Fortezza Card contains the configuration information for the Companion. The User Fortezza Card contains nine 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. Three are used to sign and verify other certificates: the local authority, the root, and the root authority certificates. The Dragonfly Companion uses the User Fortezza Card for hashing, digital signatures, key generation, and encryption. The Companion Softkey Certificate is signed at the factory with the software authority in order to prevent pirating Companions, but it is not security relevant.

A Dragonfly Companion separates two Dragonfly Domains. In general, a Dragonfly Domain is a set of computers that are networked together without any intervening Dragonfly Units. The exception is Domain 0; this is a pseudo-domain that can be specified as the domain for the local port of more than one companion (e.g., Companion PC hosts). This is described in more detail in the next section. Computers in the same domain may be PCs, Workstations, or Servers that are all at the same security level.

The Dragonfly Administration System is used to define Dragonfly Domains and their properties. Initially, there is one Dragonfly Domain. The first Dragonfly Companion (or Guard) defined creates two domains: the Local Domain and the Remote Domain. For the Companion, the PC it protects is the local Domain, and its one Ethernet interface is connected to the remote Domain. A security level is set for each Domain and these security levels may be different.

The Dragonfly Administration System is used to set the security and network configuration information. It is used to burn the information onto the Companion User Fortezza Card. The Administration System requires a Local Authority Fortezza Card to create valid Companion 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 a Dragonfly Deployment, a set of Dragonfly Domains. The Dragonfly Companion depends upon the Dragonfly Administration System to correctly configure its User Fortezza Card. The configuration can be verified anytime by the user using the Companion User Interface and by the Local Authority on the Administration System. The Dragonfly Administration System is outside the scope of this evaluation and is considered part of the environment for the Dragonfly Companion.

Dragonfly Guards and Dragonfly Companions are collectively referred to as Dragonfly Units, but the Dragonfly Guard and Dragonfly Companion are not the same. The main differences between the Companion and the Guard are as follows:

- The primary objective of the Companion is to protect the single host that it resides on from the unsecured network; the Guard is intended to protect networks of multiple hosts.
- The human user of the Companion is trusted, whereas there is no human user of the Guard.
- The local authority can configure the Companion User Fortezza Card with one or more of the options: Allow Pass Through, Firewall Mode, and Allow User to Change Default Mode. The options configured on the User Fortezza Card determine which of the following modes the user is allowed to select: Block all, Pass All, Intermediate Protection, or Firewall Protection.
- On the Companion, a user must successfully login to a Fortezza Card using the correct PIN in order to use Fortezza services. The Guard's software automatically logs in to the Fortezza card.
- The Companion does not generate proxy Address Resolution Protocol requests and responses (ARPs), whereas the Guard does.
- The Companion cannot be an Audit catcher, whereas the Guard can.

## 2.3 SECURITY SERVICES

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

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

Dragonfly Companions and Dragonfly Units 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 is formed, messages are encrypted with a symmetric key known only to the source and destination Dragonfly Companion or Guard. From a security policy perspective, the user on the Dragonfly Companion is the user operating the Dragonfly Companion host who has the User Fortezza Card. The Dragonfly Companion identifies and authenticates itself to other Dragonfly Units based on the identity associated with the User Certificate on their User Fortezza Card. The only role assumed at the Dragonfly Companion is the User Role. The user assumes the User Role when the Dragonfly Companion logs into the User Fortezza Card using the PIN provided by the user.

The Dragonfly Companion 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 Companions 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. The Dragonfly Companion 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 Companion 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 Companion 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 Companion uses bit vectors called Privilege Vectors for Discretionary Access Control (DAC) between Dragonfly Domains. Each bit, represents a Dragonfly Domain. 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 setting the appropriate bit for the domain in the Privilege Vector of the new Domain. DAC checks are performed at the time an Association is formed. When a new Companion user Fortezza Card is being configured at the Administrative System, the Local Authority can enter privileges (for other remote domains) for the local privilege vector. This enables the Companion to communicate with other hosts in those domains.

The local authority can configure a Companion so that its local side does not represent a unique domain. In such a case, the Companion is said to be a member of the pseudo domain. The pseudo domain is also known as Domain 0, because it is represented by bit 0, the first bit, in the privilege vector.

There are three ways in which the local authority can enable communications between a Companion in the pseudo domain and a host in a real domain. They are as follows:

1. Set the privilege bit for the real domain of which the host is a member in the local privilege vector of the Companion in the pseudo domain;
2. Set all the privilege bits in the local privilege vector of the Companion in the pseudo domain; or
3. Set all privilege bits in the local privilege vector of the Dragonfly Unit protecting the real domain.

There are two main differences between the pseudo domain and real domains. First, the pseudo domain is not unique. It does not meet the definition of a domain in that there are no intervening Dragonfly Units between Companions in the pseudo domain. For this reason, companions that members of the pseudo domain are also referred to as domainless companions.

Second, there is no interface for the local authority to set just bit 0 for the pseudo domain when programming the User Fortezza Card for the Dragonfly Unit protecting a real domain. Bit 0 is set only if all the bits are set. Therefore, if the local authority wants to control communications at a finer level of granularity, the local authority will have to reprogram the User Fortezza Cards for the companions in the pseudo domains, if a real domain is added later. If both domains were real domains, only the User Fortezza Card for the new Dragonfly Unit protecting the new real domain would have to programmed.

The Dragonfly Companion 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 itself and other Dragonfly Units. The Companion uses the Cipher Block Chaining CBC-64 mode of operation and the Skipjack algorithm on the User Fortezza Card.

The Dragonfly Companion checks for integrity of both User Data and Dragonfly control information when messages are transmitted between itself and other Dragonfly Units. 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 Companion can generate and send audit reports to an Audit Catcher. The Dragonfly Companion depends upon the Dragonfly Guard, which has already completed its EAL2 evaluation, to serve as its audit catcher. Audit Catchers receive audit reports from other Dragonfly Companions (and Guards) and send all messages to their serial port for printing, storage or subsequent analysis. The selection of auditable events can be controlled.

## 2.4 OPERATIONAL ENVIRONMENT

Besides providing security services, the Dragonfly Companion allows for four modes of operation. Each mode of operation is described below and any of these modes can be chosen depending on how the User Fortezza card is configured. The local authority can configure the Companion User Fortezza Card with one or more of the options: Allow Pass Through, Firewall Mode and Allow User to Change Default. Depending on what options are configured on the User Fortezza Card, the user can choose among the following modes: Block All, Pass All, Intermediate Protection and Firewall Protection.

**Block All:** This mode stops the passage of all network packets to or from the Companion host system. When the Companion User Fortezza Card is removed or when a user logs out, the Companion will default to this mode, unless the default mode has been changed to Pass All in which case the Companion will default to the Pass All mode

**Pass All:** This mode allows free network communication with all hosts and provides no security protection. In this mode, the Companion is still running, but its security features are disabled. This mode is not allowed in the evaluated configuration.

**Intermediate Protection:** This mode allows for network communication with native (i.e., non-Dragonfly) hosts in its remote domain, but uses Dragonfly encryption to communicate with other Dragonfly Units. MAC and DAC checks are performed when communicating with other Dragonfly Units. MAC checks are performed when communicating with native hosts.

**Firewall Protection:** This mode allows network communication with other Dragonfly Units only, and all the TOE security functionality is enabled..

“In addition to the modes, the No Native Associations Routing Option can be specified for a specific IP address on the routing certificate. The No Native Associations Routing option is only relevant in Intermediate Mode. When this option is set, the Companion behaves as if it were in Firewall Mode when communicating with the specified IP address.

Dragonfly Companions do not have to be programmed with complete deployment information as they use a trusted, automatic discovery mechanism to learn the system topology. Dragonfly Companions allow use of 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 Companions can exchange security levels and generate a symmetric key for encryption. Neither the initiating Dragonfly Companion nor the destination Dragonfly Unit needs to know the name, address, or even the existence of the other prior to the Association setup. Once the Association is set up, both Dragonfly Units know all that they need to know.

The Dragonfly Companion provides in-line encryption (INE) functionality to tunnel data through a network at a different security level. Dragonfly Companions 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. In addition, 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 Units exist along a data path, they provide confidentiality, integrity, and source authentication.

# 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 |
|---|---|---|
| 1 | A.Attack_Level | Attackers are assumed to have a medium level of expertise, resources, and motivation. |
| 2 | A.Crypto_Services | Cryptographic services are provided by the User Fortezza Card. |
| 3 | A.Crypto_SOF | The cryptographic algorithms on the Fortezza card are assumed strong enough to counter at least a medium level of attack. |
| 4 | A. Local_Auth | The local authority is trusted to correctly configure User Fortezza Cards. In addition, the local authority is trusted to set the time correctly on the User Fortezza Cards |
| 5 | A.No_Lower_Level_Attack | It is assumed that Windows 95 cannot be attacked through lower level network protocols (i.e., below IP layer). |
| 6 | A.No_Other_Programs | No other programs may be installed on the host computer besides Windows 95 and the Dragonfly Companion. |
| 7 | A.No_Untrusted_Users | There are no untrusted users on the Dragonfly Companion |
| 8 | A.Only_One_IP_Port | The human user is trusted to configure Windows 95 so that there is only one network and it only accepts IP datagrams. |
| 9 | A.Physical | The Dragonfly Companion Host system is assumed to be protected from physical tampering. |
| 10 | A.User | The only user on the Dragonfly Companion is the trusted human user who has been provided with the user PIN for the User Fortezza card. The human user is assumed to be able to install the Dragonfly Companion in the evaluated configuration in accordance with the IGS Procedures. The human user is assumed able to insert the correct User Fortezza Card into the Dragonfly Companion, to connect its port to the network and to put the Companion in a proper mode. The human user is trusted not to bypass or tamper with the security enforcing functions of the Dragonfly Companion. |
| 11 | A.Windows_95 | The Dragonfly Companion is installed on a Windows 95 operating system with the specified hardware configuration. (See Section 2.1.) |

**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.

| | Threat Name | Threat Description |
|---|---|---|
| 1 | T.Account | An attempted violation of the TSP may not be traceable to the Companion 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 Companion User Fortezza Card is 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.Expired | A malicious user is able to use an old User Fortezza Card or an old cryptographic key to gain unauthorized access to information. |
| 7 | T.Impersonate | An unauthorized user may attempt to impersonate a Dragonfly Companion or its trusted human user. |
| 8 | T.Inconsistent | An incorrect access control decision is made due to a security attribute being interpreted differently on another Dragonfly Unit. |
| 9 | T.Modify_Configuration | The Dragonfly Companion performs incorrectly due to either accidental or intentional modification of its configuration data by unauthorized users |
| 10 | T.Modify_Data | A message containing User or TSF Data may be modified during transmission. |
| 11 | T.No_Need_To_Know | Users have access to data that they have no need to know. |
| 12 | T.Quit | A person who is no longer an authorized user may gain access to the TOE due to a certificate not being revoked. |
| 13 | T.Sequence | It may not be possible to determine the sequence of security relevant events. |
| 14 | 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 |
| 15 | T.Tamper | A malicious user is able to interfere with the execution of the TSF software or to modify internal TSF data. |
| 16 | T.Undetected | The occurrence of a suspicious security relevant event may go undetected due to the inability to record security relevant events. |
| 17 | T.Write_Down | Information at a higher security level is released on a network at a lower security level. |
| 18 | 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.

| | Objective Name | Objective Description |
|---|---|---|
| 1 | O.Accountability | The audit data sent by the Companion to an Audit Catcher has information to identify the Companion. |
| 2 | O.Audit | The Companion must provide an audit capability that can send records of security relevant events to the Audit Catcher. |
| 3 | O.Audit_Select | The Companion must provide the ability to change the selection of auditable events during normal operation. |
| 4 | O.Authen_Source | A Companion must authenticate itself to another Dragonfly Unit. |
| 5 | O.Authen_User | A human user must authenticate her/himself to the Companion. |
| 6 | O.Confidentiality | User Data must be protected from disclosure when it is transmitted between a Companion and another Dragonfly Unit. |
| 7 | O.Consistency | TSF Data must be interpreted consistently by all the Dragonfly Units within a network. |
| 8 | O.DAC | The Companion must not release User Data to an unauthorized domain. |
| 9 | 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. |
| 10 | O.Expire | The Companion must provide the ability to check for the expiration of user certificates and keys. |
| 11 | O.Info_Flow | The Companion must not release User Data from a higher level domain to a lower level domain. |
| 12 | O.Integrity | User Data and TSF Data must be protected from modification when it is transmitted between a Companion and another Dragonfly Unit. A Companion must verify the integrity of User Data and TSF data when it is received. |
| 13 | O.Non_Bypassability | The Guard must ensure that a packet cannot be released until the security enforcing functions have been invoked and succeed. |
| 14 | O.Revoke | There must be a capability to revoke the Companion user certificates and a capability for the Companion to receive a list of revoked certificates. |
| 15 | O.Single_Level_Port | The Companion must assume that all native hosts connected to it are at the same security level as the remote port of the Companion. |
| 16 | O.SOF | The Companion must be able to meet at least a medium strength of function requirement. |
| 17 | O.Time | It must be possible to determine the time of security relevant events. |
| 18 | O.Trusted_Channel | The Companion must be able to establish a trusted communication channel between itself and another Dragonfly Unit. |

| | Objective Name | Objective Description |
|---|---|---|
| 19 | O.Verify_Config | A Companion must be able to verify that its configuration certificates have been signed by the local authority. |
| 20 | O.Windows_95 | It must be possible to configure Windows 95 so that it there is only one network port that accepts only IP datagrams. Also, it must not be possible to attack Windows 95 through network protocols below the IP layer. |

**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]

| | Objective Name | Objective Description |
|---|---|---|
| 1E | O_E.Audit_Select | The Companion must provide the ability to change the selection of auditable events during normal operation. |
| 2E | O_E.Authen_Source | A Companion must authenticate itself to another Dragonfly Unit. |
| 3E | O_E.Authen_User | A human user must authenticate her/himself to the Companion. |
| 4E | O_E.Confidentiality | User Data must be protected from disclosure when it is transmitted between a Companion and another Dragonfly Unit. |
| 5E | O_E.DAC | The Companion must not release User Data to an unauthorized domain. |
| 6E | O_E.Expire | The Companion must provide the ability to check for the expiration of user certificates and keys. |
| 7E | O_E.Info_Flow | The Companion must not release User Data from a higher level domain to a lower level domain. |
| 8E | O_E.Integrity | User Data and TSF Data must be protected from modification when it is transmitted between a Companion and another Dragonfly Unit. A Companion must verify the integrity of User Data and TSF data when it is received. |
| 9E | O_E.Revoke | There must be a capability to revoke the Companion user certificates and a capability for the Companion to receive a list of revoked certificates. |
| 10E | O_E.Single_Level_Port | All native hosts connected to the Companion must be at the same security level as the remote port of the Companion. |
| 11E | O_E.SOF | The Companion must be able to meet at least a medium strength of function requirement. |
| 12E | O_E.Time | It must be possible to determine the time of security relevant events. |
| 13E | O_E.Trusted_Channel | A Companion must be able to establish a trusted communication channel between itself and another Dragonfly Unit. |
| 14E | O_E.Verify_Config | A Companion 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.

| Objective Name | Objective Description |
|---|---|
| O-Non-IT.Local_Auth | The local authority must be adequately trained on how to configure the User Fortezza Card. |
| O-Non-IT.Physical | The Dragonfly Companion host must be protected from physical tampering. |
| O-Non-IT.Trusted_ Human_User | The trusted human user must be adequately trained to perform his/her duties in accordance with Administrator Guidance described in DF_AUM and the ADO-IGS Procedures described in DF_CUM |

**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 has 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_UAU.6 | Re-Authenticating |
| 14 | FIA_UID.2 | User identification before any action |
| **Class FMT: Security Management** | | |
| 15 | FMT_MOF.1 | Management of Security Functions Behavior |
| 16 | FMT_MTD.1 | Management of TSF Data |
| 17 | FMT_REV.1 | Revocation |
| 18 | FMT_SAE.1 | Time-limited authorization |
| 19 | FMT_SMR.1 | Security roles |
| **Class FPT: Protection of the TOE Security Functions** | | |
| 20 | FPT_ITI.1 | Inter-TSF detection of modification |
| 21 | FPT_RVM.1 | Non-bypassability of the TSP |
| 22 | FPT_SEP.1 | TSF domain separation |
| 23 | FPT_STM.1 | Reliable time stamps |
| 24 | FPT_TDC.1 | Inter-TSF basic TSF data consistency |
| **Class FTP: Trusted Path/Channels** | | |
| 25 | 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)* *Received Association Exists Message,*
*k)* *Association Granted,*
*l)* *Association Requested,*
*m)* *Association Unknown,*
*n)* *Association Type Change,*
*o)* *Audit Mask Received,*
*p)* *Bad Message Type,*
*q)* *Opening a Write Up Session,*
*r)* *Certificate or Symmetric Key Deleted,*
*s)* *Routing Table Received,*
*t)* *Save Certificate Received,*
*u)* *Routing Table Sent,*
*v)* *Internal Error,*
*w)* *Invalid Signature, ,*
*x)* *Lost Wait Queue Msg,*
*y)* *No Receipt,*
*z)* *Revoke List Received,*
*aa)* *Attempted PUD Write Down,*
*bb)* *Received by non-Audit Catcher,*
*cc)* *Release Key Unknown,*
*dd)* *Certificate Revocation List Sent,*
*ee)* *Old CRL Version,*
*ff)* *Certificate Invalid Start,*
*gg)* *Certification Expired,*
*hh)* *Certificate Revoked,*
*ii)* *Certificate Invalid,*
*jj)* *User Logs onto Companion,*
*kk)* *Mode Change,*
*ll)* *NULL Source IP Address, and*
*mm)* *Security Level Mismatch.]*

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:

a) [*event type*]

b) [*none].*

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/Companion,*

*b) object: destination domain/Companion, and*

c) *operation: release to.* ]

Dependencies: FDP_ACF.1 Security attribute based access control

Note: A Companion can be configured either as a domain or not as domain

### FDP_ACF.1 Security attribute based access control[2]

Hierarchical to: No other components.

FDP_ACF.1.1 The TSF shall enforce the [*discretionary access control SFP*] to objects based on [*privilege vectors, the four modes: Block All, Intermediate Protection and Firewall Protection and the No Native Associations Routing Option*].

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 the Dragonfly Companion is in Block All mode, all IP datagrams to and from the Companion are dropped.*

*2) If the Dragonfly Companion is in Firewall mode, and*

*a) if there are two or more Dragonfly Units between the source domain/Companion and the destination domain/Companion, then*

[2] Pass All mode is not allowed in the evaluated configuration.

*1) If the local privilege vector for the Companion or the source Domain has the bit set for the destination domain, then the IP datagram is released if the MAC check passes*

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

*3) Else, the IP datagram is not released.*

*b) or it there is only one Dragonfly Unit between the source domain/Companion and the destination domain/Companion, then the IP datagram is not released. (I.e., native mode communication is not allowed.*

*3) If the Dragonfly Companion is in Intermediate Protection mode, and*

*a) if there are two or more Dragonfly Units between the source domain/Companion and the destination domain/Companion, then*

*1) If the local privilege vector for the Companion or the source Domain has the bit set for the destination domain, then the IP datagram is released if the MAC check passes*

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

*3) Else, the IP datagram is not released.*

*b) else if there is only one Dragonfly Unit between the source domain/Companion and the destination domain/Companion, then*

*If the No Native Associations Routing Option is set for the associated IP Address,*

*Then the IP datagram is not released;*

*Else the IP datagram is released if it passes the MAC check. (I.e., native mode communication is allowed.]*

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 native host is at the same security level as the remote port of the Companion 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/Companions,*

*b)* *Information: IP datagrams,*

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

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/Companion,*

*b)* *Security level of the destination domain/Companion,*

*c)* *Type of protocol (i.e., ICMP, UDP, TCP, FTP, SMTP, or DNS),*

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

e) *Writeups enabled,* ]

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/Companion and destination domain/Companion are equal, release the IP datagram.*

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

*1)* *If writeups are disabled, no IP datagrams are released.*

*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/companion is less than the security level of the source domain/companion (writedown), the following rules apply based on the type of protocol:*

*1) If writeups are disabled, no IP datagrams are released.*

*2) If write-ups 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 the [*no additional mandatory access control SFP rules*].

FDP_IFF.2.4 The TSF shall provide the following [*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 native host is in the same security level as the remote port of the Companion 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]

### 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,*

*e)* *Routing Certificate, and*

*f)* *Cryptographic Keys]*

*Note: The trusted human user on the Dragonfly Companion is the human user operating the Dragonfly host who has the User Fortezza Card and PIN. When a Dragonfly Companion authenticates itself to another Dragonfly Unit, the user is represented by the User Certificate on the Dragonfly Companion's User Fortezza Card. These user attributes apply both to the trusted human user who has possession of the User Fortezza Card and the Dragonfly Companion.*

*The user attributes contained in the User Certificate, Configuration Certificate, Audit Certificate, Certificate Revocation List and Routing 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.*

*The companion obtains its IP address from Windows 95 from an outgoing IP datagram, rather than the IP address field in the configuration certificate.*

Dependencies: ITENV.1 Cryptographic Services on Fortezza Card

ITENV.3 Dragonfly Administration System for Setting User Attributes

ITENV.5 Certificates on the Fortezza Card

## 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

ITENV.6 Fortezza Card PINs

*Note: This requirement applies both to the Dragonfly Companion authenticating itself to other Dragonfly Units and to the human user of the Companion authenticating himself or herself by entering the User PIN for the User Fortezza Card.*

## FIA_UAU.6 Re-Authenticating

Hierarchical to: No other components.

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions [*when the User Fortezza Card is removed and re-inserted, when the User logs off and when the host is booted up*].

Dependencies: ITENV.6 Fortezza Card PINs

### 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_MOF.1 Management of Security Functions Behavior [Trusted Human User]

Hierarchical to: No other components.

FMT_MOF.1.1 The TSF shall restrict the ability to [*determine the behavior of*] the *functions [User Data Protection by setting the mode as allowed by the configuration options*] to [*the Trusted Human User*].

Dependencies: FMT_SMR.1 Security roles

ITENV.3 Dragonfly Administration System for Setting User Attributes

*Note: The options available to the user are restricted by the configuration options set by the local authority on the User Fortezza Card on the Dragonfly Administration System.*

### 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, certificate revocation list, and routing certificate*] 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 the [*Dragonfly Companion*] within the TSC to [*the local authority*].

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

*Note: The TSF provides the ability to revoke certificates that 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 unit 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, Trusted Human User*].

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

Dependencies: FIA_UID.1 Timing of identification

ITENV.3 Dragonfly Administration System for Setting User Attributes

ITENV.5 Certificates on the Fortezza Card

ITENV.6 Fortezza Card PINs

Note: Two roles, User and Trusted Human User, are used to distinguish between "the user represented by the User Certificate" and "the human user". User without modification is used for "the user represented by the User Certificate".

## 5.1.5 Class FPT: Protection of the TOE Security Functions

### 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.

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: ITENV.7 Fortezza Card Time

### 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 unit) 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 Companions only interpret TSF data from other Dragonfly Units.*

## 5.1.6 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. All messages before the establishment of an Association Request are signed.*

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

## 5.1.7 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 is 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.

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 Companion 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 Companion 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 Companion 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 Companion are set by the local authority. The user attributes are stored in the User, Configuration, Audit, Certificate Revocation, and Routing certificates on the Dragonfly Companion's User Fortezza Card.

In setting the security attributes of the Dragonfly Companion, the local authority must ensure that the level assigned to the remote port of the Companion is the same as each of the native hosts connected to that port. Each native host must be at the same level as the level of the remote port of the Companion they are attached to. The Companion assumes this to be true.

**ITENV.4 Dragonfly Administration System for Modifying TSF Data**

The Dragonfly Companion relies upon the Dragonfly Administration System to update Audit Masks, Certificate Revocation Lists, and Routing Certificates. The local authority must update the Audit Mask, Certificate Revocation List, or Routing Certificate 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 Card has to be reinitialized.

**ITENV.5 Certificates on the Fortezza Card**

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

Notes:

The user, configuration, audit, certificate revocation, and routing certificates are signed by the local authority. The local authority certificate is signed by the root, and 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 Companion TSF.

**ITENV.6 Fortezza Card PINs**

The Fortezza card requires that the correct PIN be entered before access is granted to services on the Fortezza card. The user must enter the PIN.

**ITENV.7 Fortezza Card Time**

The ITT Dragonfly Companion depends upon the time from the User Fortezza card to generate the time stored in its audit records.

# 6.0 TOE SUMMARY SPECIFICATION

## 6.1 IT SECURITY FUNCTIONS

### 6.1.1 Identification and Authentication

**IA-1. Dragonfly Companion User Fortezza Card**

A User Fortezza Card must be inserted in order for a Dragonfly Companion to start up. If the User Fortezza Card is removed, the Companion goes into either Block All or Pass All mode, depending on configuration options. The Fortezza card contains a User Fortezza Certificate that is used to identify the User Role.

**IA-2. Fortezza Card Certificate PIN**

A user must successfully login to a Fortezza Card using the correct PIN in order to use Fortezza services. The user must login every time the Companion boots, when the user logs out or when the Fortezza Card is removed and reinserted. Dragonfly Companion User Certificates and PINS for the Companion Fortezza Card 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 Companion requests an Association with another Dragonfly unit. The source Dragonfly Companion digitally signs the Association request and the destination Dragonfly Companion verifies the digital signature.

### 6.1.2 Associations

**ASSOC-1. Association as a Trusted Channel**

Dragonfly Companions form an Association that provides an Inter-TSF trusted channel between itself and other Dragonfly units. No user data is communicated until an Association is formed.

**ASSOC-2. Digitally Signed Association Request**

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

**ASSOC-3. Use of Fortezza Key Exchange Algorithm**

When a Companion forms an Association with a Dragonfly Unit, they make use of the Fortezza Key Exchange Algorithm to create a symmetric key that is known only to the Companion and the Dragonfly Unit.

**ASSOC-4. Encryption of User Data**

All user data sent between a Dragonfly Companion and a Dragonfly Unit is encrypted using the symmetric key generated by the Key Exchange Algorithm.

### 6.1.3 Discretionary Access Control (DAC)

**DAC-1. Privilege Vectors**

Dragonfly Companions 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. For the Companion, the local authority can set the local privilege vector bits for remote Domains.

**DAC-2. Shared Domain**

The local authority can configure a Companion so that its local side does not represent a unique domain. In such a case, the Companion is said to be a member of the pseudo domain. The pseudo domain is also known as Domain 0, because it is represented by bit 0, the first bit, in the privilege vector.

There are three ways in which the local authority can enable communications between a Companion in the pseudo domain and a host in a real domain. They are as follows:

1. Set the privilege bit for the real domain of which the host is a member in the local privilege vector of the Companion in the pseudo domain;
2. Set all the privilege bits in the local privilege vector of the Companion in the pseudo domain; or
3. Set all privilege bits in the local privilege vector of the Dragonfly Unit protecting the real domain.

There is no interface for the local authority to set just bit 0 for the pseudo domain when programming the User Fortezza Card for the Dragonfly Unit protecting a real domain. Bit 0 is set only if all the bits are set. Therefore, if the local authority wants to control communications at a finer level of granularity, the local authority will have to reprogram the User Fortezza Cards for the companions in the pseudo domain, if a real domain is added later.

Note for DAC-3 through DAC-5 below: A description of how the mode is configured is provided in SM-5 and SM-6.

**DAC-3. Block All Mode**

If a Dragonfly Companion is in the Block All mode, no IP datagrams are released.

**DAC-4. Intermediate Protection Mode**

If a Dragonfly Companion is in the Intermediate Protection Mode, the Companion releases all IP datagrams to native, (i.e., non-Dragonfly protected) hosts, if the MAC check passes. The Dragonfly Companion uses Dragonfly encryption to communicate with other Dragonfly Units, if the MAC and privilege vector checks pass.

**DAC-5. Firewall Protection Mode**

If the Dragonfly Companion is in Firewall mode, then IP datagrams are released to other Dragonfly Units only, if the MAC and privilege vector check pass.

**DAC-6. No Native Associations Routing Option**

If the No Native Associations Routing Option (Type field in Routing Certificate) is set, the Companions behaves as if it is in Firewall Mode, even if it is in Intermediate Mode with respect to the associated IP address. This option is only relevant in Intermediate Mode. 6.1.4 Security Levels

**SL-1. Security Levels**

Dragonfly Companions 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**

The local and remote ports on a Dragonfly Companion are both configured with security levels. The local port is configured with level of the Companion host PC. The remote port is configured with the level of the network connection. All native hosts that are connected to the Companion over a network are at the level of the remote port of the Companion.

### 6.1.5 Mandatory Access Control (MAC)

**MAC-1. Mandatory Access Control Policy.**

A Dragonfly Companion 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, ACCT, ALLO, APPE, CWD, MODE, NOOP, PASS, PORT, PWD, QUIT, STOR, STOU, STRU, TYPE, USER, and XPWD.*

**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, TOP, TURN

Other SMTP datagrams are allowed if writeups are enabled.

**MAC-5. Allowed Information Flows**

The Dragonfly Companion 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) ICMP responses

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

c) Anticipated FTP or SMTP messages as described below.

No other IP datagrams are allowed to flow from a higher level Dragonfly Domain to a lower level Dragonfly Domain.

Note: Address Resolution Protocol (ARP) and Reverse Address Resolution Protocol (RARP) requests and responses pass through the Companion without being processed by it.

**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 Companion 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 Companion releases a message for write up, it creates the anticipated response at the security level of the originating host. When the Dragonfly Companion 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 Name Server Requests and Responses**

The Dragonfly Companion releases Name Server requests and responses without performing a mandatory access control (MAC) check. Name Server 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-8. ICMP Requests and Responses**

Dragonfly Companions allow the following ICMP requests for writeup:

ICMP Echo Request, and

ICMP Time Stamp Request.

Dragonfly Companions 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.

## 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 if the security level of the remote port of the Companion is higher (if write-ups are enabled) or equal to the local port of the Companion. The Companion must be in Intermediate Protection mode to enable export of data with this policy.

**IMP-1. Import of User Data**

When User Data is imported from a Native Host, it is imported at the security level of the Dragonfly Companion's Remote port. Similarly, if from the Companion's host, the user data is imported at the security level of the Companion's local port. The Companion must be in Intermediate Protection mode to enable import of data with this policy.

## 6.1.7 Dragonfly IP Datagrams and Messages

**IP-1. Types of IP Datagrams**

From the perspective of the protection provided by the Dragonfly Companion, 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 that 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.

**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 Companion or a Dragonfly Unit. Dragonfly Pings are sent as ICMP echo request messages, one of the allowed write ups under the information flow policy.

**IP-4. Signed IP Datagrams**

Signed IP datagrams are used to transmit TSF data. Messages are digitally signed by the source Dragonfly Companion. 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 the Companion and a Dragonfly Unit. 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 Companion to another Dragonfly Unit, it is transmitted with its associated security level in a Protected User Datagram.

### 6.1.8 Confidentiality

**CONF-1. Confidentiality of User Data**

Dragonfly Companions provide confidentiality protection for User Data when it is transmitted between a Companion and a Dragonfly Unit. User Data is transmitted in a Protected User Datagram (PUD) that is encrypted with a symmetric key known only to the Companion and the destination Dragonfly unit.

### 6.1.9 Integrity

**INT-1. Integrity of User Data**

User Data is always transmitted between a Companion and a Dragonfly Unit 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 the Companion and the destination Dragonfly Unit. In the case of encrypted checksums, a symmetric key (known only to the Companion and the destination Dragonfly Unit) is used to encrypt the data and the checksum. If another Dragonfly unit exists between the Companion and destination Dragonfly Unit, a second symmetric key (known only to adjacent Dragonfly unit) is used to encrypt a second checksum so that the intermediate Dragonfly unit can verify the integrity of the message without decrypting it.

### 6.1.10 Audit

**AUDIT-1: Audit Catcher**

The Dragonfly Companion sends its audit messages to a Dragonfly Guard serving as audit catcher, also known as an audit catcher.

**AUDIT-2: Audit Required Configuration Option**

The local authority specifies whether or not audit will be required when the Companion User Fortezza Card is configured on the Dragonfly Administration System. If audit is required, the Dragonfly Companion 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 Companion User Fortezza Card is configured on the Dragonfly Administration System. The Dragonfly Companion 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 Companion stops processing.

**AUDIT-4. Audit Catcher Messages**

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

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

Tables 6.1 through 6.5 depict the message formats.

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

Audit Event Messages are sent from Dragonfly Companions to Audit Catchers to report an auditable event. The IP address in the Audit Event Message is the same IP address as the one in the Check-In message. The IP address comes from Windows 95 from an outgoing IP datagram, not the configuration certificate.

The IP address on Windows 95 cannot be changed without restarting Windows 95 and thus, restarting the companion.

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

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

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

| | | |
|---|---|---|
| IP HEADER | | |
| UDP HHEADER | | |
| ENCRYPTION IV | | |
| ORIGINAL IP HEADER | | |
| MESSAGE NUMBER | | REVOKE LIST VERSION |
| SENDING DATE / TIME | | |
| SOFTWARE VERSION | | |
| AUDIT MASK NAME | | |
| AUDIT MASK VERSION | | SPARE |
| STATUS COUNTERS | | |
| TYPE : 0x35 | FMT: 0x32 | DF FLAG |
| Spare | Padding to 8-Byte Boundary of Encrypted Data | |
| Type 1 MSG AUTHENTICATION | | |
| Association ID | | |
| TYPE: *0x0* | FMT: *0x30* | DF FLAG: *0xdfdf* |

**Table 6.1 - Initialization/Check-In Message**

| | | |
|---|---|---|
| IP HEADER | | |
| UDP HEADER | | |
| ENCRYPTION IV | | |
| ORIGINAL IP HEADER | | |
| A-E MESSAGE NUMBER | | AUDIT EVENT CODE |
| SENDING DATE / TIME | | |
| AUDIT EVENT TEXT | | |
| TYPE: 0x33 | FMT: 0x32 | DF FLAG |
| Spare | PADDING TO 8-BYTE BOUNDARY of Encrypted Data | |
| Type 1 MSG AUTHENTICATION | | |
| Association ID | | |
| TYPE: *0x0* | FMT: *0x30* | DF FLAG: *0xdfdf* |

**Table 6.2 - Audit Event Message**

| IP HEADER | | |
|---|---|---|
| UDP HEADER | | |
| ENCRYPTION IV ... | | |
| ORIGINAL IP HEADER | | |
| MESSAGE NUMBER | | SPARE |
| SENDING TIME / DATE | | |
| CERTIFICATE TYPE | | CERTIFICATE LENGTH |
| ISSUER DISTINGUISHED NAME | | |
| AUDIT MASK VERSION | | SPARE |
| EXPIRE TIME | | |
| AUDIT MASK | | |
| AUDIT MASK NAME | | |
| SIGNATURE (CERTIFICATE) | | |
| MSG: 0x36 | FMT: 0x32 | DF FLAG |
| Spare | PADDING TO 8-BYTE BOUNDARY of Encrypted Data | |
| Type 1 MSG AUTHENTICATION | | |
| Association ID | | |
| TYPE: *0x0* | FMT: *0x30* | DF FLAG: *0xdfdf* |

**Table 6.3 - Audit Mask Message**

| IP HEADER | | |
|---|---|---|
| UDP HEADER | | |
| ENCRYPTION IV | | |
| ORIGINAL IP HEADER | | |
| MESSAGE NUMBER | | SPARE |
| SENDING TIME / DATE | | |
| CERTIFICATE TYPE | | CERTIFICATE LENGTH |
| ISSUER DISTINGUISHED NAME | | |
| REVOKE LIST ID NUMBER | | REVOKED ID COUNT |
| EXPIRE TIME | | |
| REVOKED CERTIFICATE ID NUMBERS | | |
| SIGNATURE (Certificate) | | |
| MSG: 0x3E | FMT: 0x32 | DF FLAG |
| Spare | PADDING TO 8-BYTE BOUNDARY of Encrypted Data | |
| Type 1 MSG AUTHENTICATION | | |
| Association ID | | |
| TYPE: *0x0* | FMT: *0x32* | DF FLAG: *0xdfdf* |

**Table 6.4 - Revocation Message**

| | | | |
|---|---|---|---|
| IP HEADER | | | |
| UDP HEADER | | | |
| ENCRYPTION IV | | | |
| ORIGINAL IP HEADER | | | |
| MESSAGE NUMBER | | ORIG MSG TYPE | HOLD FLAG |
| SENDING DATE / TIME | | | |
| TYPE: 0x3D | FMT: 0x32 | DF FLAG | |
| Spare | PADDING TO 8-BYTE BOUNDARY of Encrypted Data | | |
| PADDING continued … | | Type 1 MSG AUTHENTICATION | |
| Association ID | | | |
| TYPE: *0x0* | FMT: *0x30* | DF FLAG: *0xdfdf* | |

**Table 6.5 - Receipt Message**

**AUDIT-5: Audit User Interface**

The Companion provides a user interface to check the Audit information. Specifically the Companion has a view/log menu bar option that displays the Companion Log and Events Log. The Companion Log contains the settings information generated during startup and initialization. The Events Log contains detailed information regarding the sending of audit records.

**AUDIT-6: Auditable Events**

Table 6.6 lists Audit Event Codes and their corresponding Event Name and Description.

| No. | Event Name | Event Description |
|---|---|---|
| | Audit Startup | Check-in message from a Companion to its audit catcher;<br>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.) |

| No. | Event Name | Event Description |
|---|---|---|
| 8 | Association Closed | Reporting Dragonfly Unit detects that an Association has been closed because it has timed out, its Certificate expired or was revoked. |
| 9 | Received Association Exists Message | This is inherent in the normal recovery process. There is no security implication. The event is reported by the Dragonfly Unit that receives the Association Request when the Association already exists. |
| 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 the recycling of a Dragonfly Unit and has no security impact. |
| 13 | Association Type Change | A message has been received directly from a host after a non-native association had been established. Usually indicates a Companion has changed to Pass-All mode. In Pass-all mode the Companion can not talk to an audit catcher. |
| 14 | Audit Catcher List Received. | Not implemented. |
| 15 | Audit Mask Received | The Audit Mask was received. |
| 16 | Bad Message Type. | Dragonfly Unit has received a Dragonfly message that is formatted incorrectly. |
| 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 | Routing Table Received | A message containing a routing certificate has been received |
| 20 | Save Certificate Received | A certificate could not be saved on the Fortezza card. |
| 21 | Routing Table Sent | A message containing a routing certificate has been sent. |
| 22 | Internal Error | Software error. |
| 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 receives an Association Grant or Deny Message and could not find the Association request. Relevant only in Intermediate Dragonfly units and not applicable to Companions |
| 26 | No Receipt | Dragonfly Unit did not receive a receipt for a non-audit message. Examples involve the Audit Catcher reporting that a receipt was not received for an Audit Mask or CRT (Certificate Revocation List message). |
| 27 | Revoke List Received | Dragonfly Unit has received an updated Revocation List. |
| 28 | Attempted PUD Write Down | An attempt was made to Write Down User Data. |
| 29 | Received by non-Audit Catcher | A non-Audit Catcher received a message that should have been sent to an Audit Catcher. |
| 30 | Release Key Unknown | An Intermediate Guard received a PUD and cannot find the release key corresponding to that message. The message is thrown away and is not released. |

| No. | Event Name | Event Description |
|---|---|---|
| 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. Currently, this is applicable only to Guards that are acting as Audit Catchers. |
| 33 | Old CRL Version | The Audit Catcher has received a Check In Message referencing an out of date CRL. |
| 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 | The user has logged onto the Companion |
| 39 | User Logs off Companion | Not implemented because the Fortezza queue is not processed after log off. |
| 40 | Companion changes mode | The Companion mode has been changed. |
| 41 | Audit Catcher Unreachable | Dragonfly Companion has entered Pass All Mode. Not applicable to Dragonfly Companion in evaluated configuration or Dragonfly Guard. |
| 42 | NULL Source IP Address | Reporting DF Unit has received a native message with a NULL Source IP Address. |
| 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 6.6 – 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 Companion, it can select either Standard, Audit All, or Audit None.

If the Dragonfly Companion 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 Companion's initial configuration and are updated by the Audit Catcher. The Audit Mask is identified by name and version number.

The Dragonfly Companion 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 Companion has an out-of-date Audit Mask, the Audit Catcher sends the current Audit Mask back to the Dragonfly Companion.

Note: Audit Mask messages cannot be sent from an Audit Catcher to a Dragonfly Companion until that Dragonfly Companion checks in with the Audit Catcher and the Audit Mask version is updated. If the check in period is very long, the Companion could miss the auditing of some new events if they occurred while the Audit Catcher was waiting for the Companion 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.

**Audit-9. Audit Catcher**

The Dragonfly Companion depends upon a Dragonfly Guard configured to serve as an audit catcher to receive the audit records that it generates. The audit catcher is also required to send updated certificate revocation lists, audit masks, and routing certificates to the Dragonfly Companion.

Any Dragonfly Guard can be specified as an Audit Catcher. An Audit catcher receives Audit messages from Dragonfly Units 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.

**Audit-10. Audit Report**

The Audit output is in ASCII format. An Audit Report out of the audit catcher contains the following fields:

- Companion 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.

## 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 Companion 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 Dragonfly Units 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 Units check in with the Audit Catcher, the Audit Catcher sends them the new CRL, if the new CRL is more recent than the Companion's current CRL. Dragonfly Companions will then reject packets originating from Dragonfly Units 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 comes from the time on the local authority workstation. This time can be set by the Local Authority before he configures the Companion User Fortezza Card. The time on the Companion User Fortezza Card is taken from the local authority workstation when the Companion User Fortezza Card is configured.

**TIME-2. Companion Time**

The Companion reads the time from its User Fortezza Card and uses it to compute the time to be stored in its audit records and for CRL processing.

### 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, and Certificate Revocation Certificates. The content of these certificates is shown in Tables 6.7 through 6.11. The security attributes stored in the certificates on the Fortezza card are set by the local authority on the Dragonfly Administration System. Depending on the settings of the mode-related configuration options (see Table 6.12 under SM-5), the trusted human user may update the mode.

| **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 6.7 – Contents of User Certificate**

| **Configuration Certificate** |
| --- |
| Certificate Type |
| Certificate Length |
| Issuer  Distinguished Name |
| **Configuration Files** |
| Configuration File Version |
| ***[Local Port]*** |
| Security Level |
| Firewall Protection |
| ***[ Remote Port ]*** |
| Security Level |
| Firewall Protection |
| ***[Timing]*** |
| Wait for Receipt Delay |
| Wait for Association Delay |
| Receipt Retries |
| Association Time to Live |
| Association Check Period |
| Crypto Period |

| **Configuration Certificate (Continued)** |
|---|
| ***[Audit]*** |
| Audit Catcher_Required |
| Checkin_Period |
| Enable Anticipated Messages |
| 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*] |
| ***[SNIU Configuration]*** |
| MSE Port (=neither) |
| SNIU Name |
| Allow_Pass_Through |
| Allow_Default_Pt |
| Authority_Port |
| **Signature** |

**Table 6.8 – Contents of Configuration Certificate**

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

**Table 6.9 – Contents of Audit Mask 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 6.10 – 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 |
| [Items IP Address through Type may be repeated] |
| Signature |

**Table 6.11 – 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 Companion Fortezza card has the following nine 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,

h) Routing signed by Local Authority, and

i) Companion Softkey signed by ITT Software Authority.

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

Note: the Companion Softkey certificate is not security relevant. (The information in these certificates is not used to implement any of the TOE Security Functionality described in this ST.)

**SM-2. Dragonfly Administration System**

The User Fortezza Card for a Dragonfly Companion is configured by the local authority on the Dragonfly Administration System. The User can view the configuration of the User Fortezza card using the local interface provided by the Companion and the local authority can check the configuration of the card on the Dragonfly Administration System.

**SM-3 Management of TSF Data**

Initially, a Companion uses the Audit Mask, Certificate Revocation List, and Routing Certificate stored on its own User Fortezza Card. A Dragonfly Companion's Audit Mask, Certificate Revocation List, and Routing Certificate 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 Companion, it will send it an Audit Mask Message if the Companion's Audit Mask is out of date, a Revocation Message if the Companion's Certificate Revocation List is out of date, or a Routing Table Message if the Routing Certificate is out of date.

**SM-4 Configuration Options for Mode**

There are three configuration options related to mode on the User Fortezza Card that can be set by the local authority on the Dragonfly Administration System:

a) Firewall Mode option,

b) Pass Through Allowed option, and

c) Allow User to Change Default option.

Pass Through Allowed and Allow User to Change Default must be disabled in the evaluated configuration. The behavior of these options is described below.

**Firewall Mode Option:**

This option only has effect on the Companion when a user is logged in to the Companion. If the local authority selects this option, the trusted human user of the Companion is not able to select Intermediate Protection Mode, but only Firewall Protection Mode.

**Pass Though Allowed Option:**

If the local authority checks this parameter the Companion defaults to Pass All mode when the trusted human user is not logged in and the trusted human user of the Companion is able to select Pass All Mode. If this option is not selected by the local authority, the default mode when a companion user is not logged in is Block All mode. Pass All mode is not selectable by the trusted human user, unless the Allow User to Change Default option is set. Because this option causes the Companion to default to Pass All mode when not logged in, this option must not be selected in the evaluated configuration. The Dragonfly Administration System User Manual directs the local authority to disable this option in the evaluated configuration.

**Allow User to Change Default Option:**

If the local authority does not select this option, the trusted human user will not be able to change the default mode of the Companion. If the local authority does select this option, the trusted human user will be able to change the default mode to Pass All, even if the card does not have it set as its default value. In other words, if the local authority does not set Pass Through Allowed option but does set this bit, the trusted human user could set the default mode before login to Pass All mode. The Dragonfly Administration System User Manual directs the local authority to disable this option in the evaluated configuration.

**SM-5. Allowable Modes**

The allowable modes are determined by the configuration options as shown in Table 6.12. The first half of the table shows the mode that the Companion can default to if the trusted human user is not logged in. The second half of the table shows the modes that the trusted human user can select when logged in.

There are three ways in which the trusted human user can log off the companion:

1) By selecting the Log Off option

2) By shutting down the Companion user interface, and

3) By removing the Fortezza card.

When logoff occurs, the Companion will revert to its default state of either Block All mode or Pass All mode. In the evaluated configuration, this will always be Block All mode. When the user logs off, the values for the three configuration options: Pass Through Allowed, Firewall Mode, and Allow User to Change Default are stored in the Windows 95 registry and used until a user logs in again. Only Firewall Mode can be changed in the evaluated configuration. Pass Through Allowed and Allow User to Change Default are always disabled in the evaluated configuration.

If the trusted human user is not logged in, the only allowed mode is Block All mode, if the Pass Through Allowed option is not set on the Companion User Fortezza card as required in the evaluated configuration. The Companion allows Block All and Pass All mode, if the Pass Through Allowed option is configured on the Companion User Fortezza card. If the card is set with the Allow User to Change Default option, the user can set the default to Pass All mode using the Pass All Packets Before Login Entry. The Pass Through Allowed option is also disabled in the evaluated configuration.

If the trusted human user is logged in, the Companion can always be used in Block All mode or Firewall Protection mode. The Companion allows Pass All mode, if the Companion is configured with the Pass

Through Allowed option, The Companion allows Intermediate Protection mode, only if the Firewall Mode option is not configured on the User Fortezza Card.

| **Logged In** | **Pass Through Allowed**** | **Firewall Mode** | **Allow User to Change Default **** | **Allowed Modes 1=Block All 2=Pass All 3=Intermed. Protection 4=Firewall Protection** | | | |
|---|---|---|---|---|---|---|---|
| No*** | No | No | No | **1*** | | | |
| No++ | No | No | Yes | **1*** | 2+ | | |
| No | No | Yes | No | **1*** | | | |
| No++ | No | Yes | Yes | **1*** | 2+ | | |
| No++ | Yes | No | No | 1 | **2*** | | |
| No++ | Yes | No | Yes | 1 | **2*** | | |
| No++ | Yes | Yes | No | 1 | **2*** | | |
| No++ | Yes | Yes | Yes | 1 | **2*** | | |
| Yes | No | No | No | 1 | | **3*** | 4 |
| Yes++ | No | No | Yes | 1 | | **3*** | 4 |
| Yes | No | Yes | No | 1 | | | **4*** |
| Yes++ | No | Yes | Yes | 1 | | | **4*** |
| Yes++ | Yes | No | No | 1 | 2 | **3*** | 4 |
| Yes++ | Yes | No | Yes | 1 | 2 | **3*** | 4 |
| Yes++ | Yes | Yes | No | 1 | 2 | | **4*** |
| Yes++ | Yes | Yes | Yes | 1 | 2 | | **4*** |

* The **bold** values in the table are the modes that the software will go to by default without user reconfiguration.

** Option not allowed in the evaluated configuration.

*** This is the initial state of the Dragonfly Companion after installation.

+ The trusted human user can set the default mode to Pass All mode by making use of the Pass All Packets Before Login Entry if the Allow User to Change Default option is enabled. This state will not be reached in the evaluated configuration.

++ This state will not be reached in the evaluated configuration, because Pass Through Allowed and Allow User to Change Default must always be set to "no."

**Table 6.12 - Modes Allowed by Configuration Options**

**SM-6 Mode Set by Trusted Human User**

The security state menu of the Dragonfly Companion has four options: Block All, Pass All, Intermediate Protection, and Firewall Protection. The trusted human user can select one of these options when logged in, based on the rules depicted in Table 6.11.

If the Allow User to Change Default option is set to "Yes", the Pass All Packets Before Login Menu Entry is enabled, and the trusted human user is allowed to change the default mode before login to Pass All.

#### SM-7 Reaching Modes

**Block All Mode**: Block All mode is the default mode when the trusted human user is not logged in, unless the Pass Through Allowed option has been selected by the local authority which is not allowed in the evaluated configuration. Block All mode can always be selected by the trusted human user when s/he is logged in.

**Pass All Mode**: Pass All is the default mode when the trusted human user is not logged in, if the Pass Through Allowed option is selected. However, Pass All mode is not allowed in the evaluated configuration. This option also allows the trusted human user to select Pass All mode when logged in. If the Allow User to Change Default option is selected, the trusted human user has access to the Pass All Packets Before Login Entry and can use this to set the default mode to Pass All.

**Intermediate Protection Mode:** The Companion defaults to Intermediate Protection mode when the trusted human user is logged in, if the Firewall Mode option is not set. If the Companion has been set to some other mode, the trusted human user can set it back to Intermediate Protection Mode when logged in, if the Firewall Mode option is not set.

**Firewall Protection Mode:** The Companion defaults to Firewall Protection mode when the trusted human user is logged in, if the Firewall Mode option is set. If the Companion has been set to some other mode, the trusted human user can set it back to Firewall Protection Mode when logged in.

### 6.1.15 Inter-TSF Basic Data Consistency

#### CONS-1. Inter-TSF Data Consistency

Dragonfly Companions inter-operate with other Dragonfly Units (Companions and Guards). Most security-relevant values such as security levels and audit masks are constants that are the same on all Dragonfly Units. 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 Companion performs access control checks on all incoming and outgoing IP datagrams. The Dragonfly Companion Driver is placed between the Transport Driver Interface (TDI) MS TCP/IP Protocol Driver and the Network Card Interface (NIC) Ethernet Card Driver.

#### SA-2. TSF Domain Separation

The Dragonfly Companion maintains a security domain for its own execution that protects it from interference and tampering by untrusted subjects. Since no untrusted human users or untrusted software is allowed on the Companion host PC, the only interface available to untrusted users is the network interface. Windows 95 is configured to accept only IP datagrams. The ITT Dragonfly Companion processes all incoming IP datagrams. Lower level protocol messages (i.e., below the IP layer) are not processed by the Dragonfly Companion.

#### SA-3. Windows 95

Windows 95 can be configured so that it has only one network port and only accepts IP datagrams. Windows 95 cannot be attacked by lower level network protocols (i.e., below the IP layer).

The ITT Dragonfly Companion gets its IP address from Windows 95 by waiting for an outgoing IP message and picking up its IP address from it. The Windows 95 IP address cannot be changed without restarting Windows 95 and thus restarting the Companion. Therefore, the IP address in the Initialization/Check-In Message is the same as the IP address in the Audit Event Message.

## 6.2 ASSURANCE MEASURES

The Dragonfly Companion claims to satisfy the assurance requirements for Evaluation Assurance Level EAL2. The following items were provided by the developer as evaluation evidence to satisfy the EAL2 Assurance Requirements:

a) Configuration Management (CM) Documentation,

b) Delivery Procedures

c) Functional Specification,

d) High-Level Design,

e) Representation Correspondence,

f) Administrator Guidance,

g) User Guidance,

h) Test Coverage Analysis,

i) Test Documentation,

j) TOE for Testing,

k) SOF Analysis, and

l) Vulnerability Analysis

Table 8.14 – 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.

| | **Threat Name** | **Threat Description** | **Objective** |
|---|---|---|---|
| 1 | T.Account | An attempted violation of the TSP may not be traceable to the Companion where it occurred. | O.Accountability<br>O.Audit |
| 2 | T.Acquire_Key | An unauthorized user is able to acquire the key for an encrypted message. | O.Trusted_Channel<br>O_E.Trusted_Channel |
| 3 | T.Bypass | A user is able to bypass the security enforcing functions | O.Non-Bypassability<br>O.Windows_95 |
| 4 | T.Card_Lost | The Companion User Fortezza Card is lost and recovered by a malicious user. | O.Revoke<br>O_E.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<br>O_E.Confidentiality |
| 6 | 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<br>O_E.Expire |
| 7 | T.Impersonate | An unauthorized user may attempt to impersonate a Dragonfly Companion or its trusted human user. | O.Authen_Source<br>O.Authen_User<br>O.Trusted_Channel<br>O_E.Authen_Source<br>O_E.Authen_User<br>O_E.Trusted_Channel |
| 8 | T.Inconsistent | An incorrect access control decision is made due to a security attributes being interpreted differently on another Dragonfly Unit. | O.Consistency |
| 9 | T.Modify_Configuration | The Dragonfly Companion performs incorrectly due to either accidental or intentional modification of its configuration data by unauthorized users. | O.Verify_Config<br>O_E.Verify_Config |
| 10 | T.Modify_Data | A message containing User or TSF Data may be modified during transmission. | O.Integrity<br>O_E.Integrity |
| 11 | T.No_Need_To_Know | Users have access to data that they have no need to know. | O.DAC<br>O_E.DAC |

| | Threat Name | Threat Description | Objective |
|---|---|---|---|
| 12 | T.Quit | A person who is no longer an authorized user may gain access to the TOE due to a certificate not being revoked. | O.Revoke<br>O_E.Revoke |
| 13 | T.Sequence | It may not be possible to determine the sequence of security relevant events. | O.Time<br>O_E.Time |
| 14 | 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<br>O_E.Audit_Select |
| 15 | T.Tamper | A malicious user is able to interfere with the execution of the TSF software or modify internal TSF data. | O.Domain_Separation<br>O.Windows_95 |
| 16 | T.Undetected | The occurrence of a suspicious security relevant event may go undetected due to the inability to record security relevant events. | O.Audit |
| 17 | T.Write_Down | Information at a higher security level is released on a network at a lower security level. | O.Info_Flow<br>O_E.Info_Flow |
| 18 | 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<br>O.Single_Level_Port<br>O_E.Info_Flow<br>O_E.Single_Level_ Port |

**Table 8.1 – All Threats to Security Addressed by Objectives**

Table 8.2 identifies the objectives that address each organizational security policy.

| Policy Name | Organizational Security Policy | Objective |
|---|---|---|
| P.Audit | It must be possible to record security relevant actions. | O.Accountability<br>O.Audit<br>O.Audit_Select |
| 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.Info_Flow |

**Table 8.2 – All Organizational Security Policies Met by Objectives**

Table 8.3 identifies the objectives that address each of the secure usage assumptions. Objectives prefixed by O. only are objectives for the TOE. Objectives prefixed by O_E. are objectives for the IT environment. Objectives prefixed by Non-IT are Non-IT objectives for the environment.

| | Assumption Name | Assumption Description | Objective |
|---|---|---|---|
| 1 | A.Attack_Level | Attackers are assumed to have a medium level of expertise, resources, and motivation. | O.SOF<br>O_E.SOF |
| 2 | A.Crypto_Services | Cryptographic services are provided by the User Fortezza Card. | O.SOF<br>O_E.SOF |
| 3 | A.Crypto_SOF | The cryptographic algorithms on the Fortezza card are assumed strong enough to counter at least a medium level of attack. | O.SOF<br>O_E.SOF |
| 4 | A. Local_Auth | The local authority is trusted to correctly configure User Fortezza Cards. | O.Non-IT.Local_Auth |
| 5 | A.No_Lower_Level_Attack | It is assumed that Windows 95 cannot be attacked through lower level network protocols (i.e., below IP layer.) | O.Windows_95 |
| 6 | A.No_Other_Programs | No other programs may be installed on the host computer besides Windows 95 and the Dragonfly Companion. | O.Non-IT.Trusted_<br>_Human_User |
| 7 | A.No_Untrusted_Users | There are no untrusted users on the Dragonfly Companion | O.Non-IT.Trusted_<br>_Human_User |
| 8 | A.Only_One_IP_Port | The human user is trusted to configure Windows 95 so that there is only one network and it only accepts IP datagrams. | O.Non-IT.Trusted_<br>_Human_User<br>O.Windows_95 |
| 9 | A.Physical | The Dragonfly Companion Host system is assumed to be protected from physical tampering. | O.Non-IT.Physical |
| 10 | A. User | The only user on the Dragonfly Companion is the trusted human user who has been provided with the user PIN for the User Fortezza card. The human user is assumed to be able to install the Dragonfly Companion in the evaluated configuration in accordance with the IGS Procedures. The human user is assumed able to insert the correct User Fortezza Card into the Dragonfly Companion, to connect its port to the network and to put the Companion in a proper mode. The human user is trusted not to bypass or tamper with the security enforcing functions of the Dragonfly Companion. | O.Non-IT.Trusted_<br>Human_User |
| 11 | A.Windows_95 | The Dragonfly Companion is installed on a Windows 95 operating system with the specified hardware configuration. | O.Non-IT.Trusted_<br>Human_User |

**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 for the TOE, since each objective addresses at least one threat, organizational security policy, or secure usage assumption.

| | **Objective Name** | **Objective Description** | **Threat/Policy/ Assumption** |
|---|---|---|---|
| 1 | O.Accountability | The audit data sent by the Companion to an Audit Catcher has information to identify the Companion. | T.Account |
| 2 | O.Audit | The Companion must provide an audit capability that can send records of security relevant events to the Audit Catcher. | T.Undetected<br>P.Audit |
| 3 | O.Audit_Select | The Companion must provide the ability to change the selection of auditable events during normal operation. | T.Static_Audit |
| 4 | O.Authen_Source | A Companion must authenticate itself to another Dragonfly Unit. | T.Impersonate |
| 5 | O.Authen_User | A human user must authenticate her/himself to the Companion. | T.Impersonate |
| 6 | O.Confidentiality | User Data must be protected from disclosure when it is transmitted between a Companion and another Dragonfly Unit. | T.Confidential |
| 7 | O.Consistency | TSF Data must be interpreted consistently by all the Dragonfly Units within a network. | T.Inconsistent |
| 8 | O.DAC | The Companion must not release User Data to an unauthorized domain. | T.No_Need_To_Know<br>P.DAC |
| 9 | 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 |
| 10 | O.Expire | The Companion must provide the ability to check for the expiration of user certificates and keys. | T.Expired |
| 11 | O.Info_Flow | The Companion must not release User Data from a higher level domain to a lower level domain. | T.Write_Down<br>T.Wrong_Level<br>P.MAC |
| 12 | O.Integrity | User Data and TSF Data must be protected from modification when it is transmitted between a Companion and another Dragonfly Unit. A Companion must verify the integrity of User Data and TSF data when it is received. | T.Modify_Data |
| 13 | 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 |
| 14 | O.Revoke | There must be a capability to revoke the Companion user certificates and a capability for the Companion to receive a list of revoked certificates. | T.Card_Lost<br>T.Quit |
| 15 | O.Single_Level_Port | The Companion must assume that all native hosts connected to it are at the same security level as the remote port of the Companion. | T.Wrong_Level |
| 16 | O.SOF | The Companion must be able to meet at least a medium strength of function requirement. | A.Attack_Level<br>A.Crypto_Services<br>A.Crypto_SOF |

| | Objective Name | Objective Description | Threat/Policy/ Assumption |
|---|---|---|---|
| 17 | O.Time | It must be possible to determine the time of security relevant events. | T.Sequence |
| 18 | O.Trusted_Channel | The Companion must be able to establish a trusted communication channel between itself and another Dragonfly Unit. | T.Acquire_Key<br>T.Impersonate |
| 19 | O.Verify_Config | A Companion must be able to verify that its configuration certificates have been signed by the local authority. | T.Modify_Configura-tion |
| 20 | O.Windows_95 | It must be possible to configure Windows 95 so that it there is only one network port that accepts only IP datagrams. Also, it must not be possible to attack Windows 95 through network protocols below the IP layer. | A.No_Lower_Level_ Attack<br>A.Only_One_IP_Port |

**Table 8.4 – All IT Security Objectives for the TOE Necessary**

Table 8.5 shows that there are no unnecessary IT security objectives for the Environment, since each objective addresses at least one threat, organizational security policy, or secure usage assumption.

| | **Objective Name** | **Objective Description** | **Threat/Policy/ Assumption** |
|---|---|---|---|
| 1E | O_E.Audit_Select | The Companion must provide the ability to change the selection of auditable events during normal operation. | T.Static_Audit<br>P.Audit |
| 2E | O_E.Authen_Source | A Companion must authenticate itself to another Dragonfly Unit. | T.Impersonate |
| 3E | O_E.Authen_User | A human user must authenticate her/himself to the Companion. | T.Impersonate |
| 4E | O_E.Confidentiality | User Data must be protected from disclosure when it is transmitted between a Companion and another Dragonfly Unit. | T.Confidential |
| 5E | O_E.DAC | The Companion must not release User Data to an unauthorized domain. | T.No_Need_To_Know<br>P.DAC |
| 6 | O_E.Expire | The Companion must provide the ability to check for the expiration of user certificates and keys. | T.Expired |
| 7E | O_E.Info_Flow | The Companion must not release User Data from a higher level domain to a lower level domain. | T.Write_Down<br>P.MAC |
| 8E | O_E.Integrity | User Data and TSF Data must be protected from modification when it is transmitted between a Companion and another Dragonfly Unit. A Companion must verify the integrity of User Data and TSF data when it is received. | T.Modify_Data |
| 9E | O_E.Revoke | There must be a capability to revoke the Companion user certificates and a capability for the Companion to receive a list of revoked certificates. | T.Card_Lost<br>T.Quit |
| 10E | O_E.Single_Level_Port | All native hosts connected to the Companion must be at the same security level as the remote port of the Companion. | T.Wrong_Level |
| 11E | O_E.SOF | The Companion must be able to meet at least a medium strength of function requirement. | A.Attack_Level<br>A.Crypto_Services<br>A.Crypto_SOF |
| 12E | O_E.Time | It must be possible to determine the time of security relevant events | T.Sequence |
| 13E | O_E.Trusted_Channel | A Companion must be able to establish a trusted communication channel between itself and another Dragonfly Unit. | T.Acquire_Key<br>T.Impersonate |
| 14E | O_E.Verify_Config | A Companion must be able to verify that its configuration certificates have been signed by the local authority. | T.Modify_Configuration |

**Table 8.5 – All IT Security Objectives for the Environment Necessary**

Table 8.6 shows that there are no unnecessary Non-IT objectives.

| Objective Name | Objective Description | Assumption |
|---|---|---|
| O-Non-IT.Local _Auth | The local authority must be adequately trained on how to configure the User Fortezza Card. | A.Local_Auth |
| O-Non-IT.Physical | The Dragonfly Companion host must be protected from physical tampering. | A.Physical |
| O-Non-IT.Trusted_ Human_User | The trusted human user must be adequately trained to perform his/her duties in accordance with Administrator Guidance and the IGS Procedures | A.No_Other_Programs<br>A.No_Untrusted_Users<br>A.Only_One_IP_Port<br>A.User<br>A.Windows_95 |

**Table 8.6 – All Non-IT Security Objectives Necessary**

## 8.2 SECURITY REQUIREMENTS RATIONALE

### 8.2.1 All Objectives Met by Security Requirements

Tables 8.7 and 8.8 show 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, the Dragonfly Guard serving as an Audit Catcher, and/or the User Fortezza Card.) Since the Common Criteria 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 as shown in Table 8.7. Security Objectives for the Environment are satisfied by IT requirements for the environment (ITENV.n) as shown in Table 8.8.

| No | Objective Name | Security Requirement |
|---|---|---|
| 1 | O.Accountability | FAU_GEN.1 |
| 2 | O.Audit | FAU_GEN.1 |
| 3 | O.Audit_Select | FAU_SEL.1<br>FMT_MOF.1<br>FMT_MTD.1 |
| 4 | O.Authen_Source | FIA_ATD.1<br>FIA_UAU.2<br>FIA_UID.2 |
| 5 | O.Authen_User | FIA_UAU.2<br>FIA_UAU.6<br>FIA_UID.2 |
| 6 | O.Confidentiality | FDP_UCT.1 |
| 7 | O.Consistency | FPT_TDC.1 |
| 8 | O.DAC | FDP_ACC.1<br>FDP_ACF.1<br>FIA_ATD.1 |
| 9 | O.Domain_Separation | FPT_SEP.1 |
| 10 | O.Expire | FMT_SAE.1 |
| 11 | O.Info_Flow | FDP_IFC.1<br>FDP_IFF.2<br>FIA_ATD.1 |
| 12 | O.Integrity | FDP_UIT.1<br>FPT_ITI.1 |
| 13 | O.Non_Bypassability | FPT_RVM.1 |
| 14 | O.Revoke | FMT_REV.1<br>FMT_MOF.1<br>FMT_MTD.1 |
| 15 | O.Single_Level_Port | FDP_ETC.1<br>FDP_ITC.1 |
| 16 | O.SOF | FDP_UIT.1 |
| 17 | O.Time | FPT_STM.1 |
| 18 | O.Trusted_Channel | FTP_ITC.1 |
| 19 | O.Verify_Config | FMT_SMR.1 |
| 20 | O.Windows_95 | FPT_RVM.1<br>FPT_SEP.1 |

**Table 8.7 – Mapping of IT Security Objectives to Functional Requirements**

| No | Objective Name | Security Requirement |
|---|---|---|
| 1E | O_E.Audit_Select | ITENV.3<br>ITENV.4 |
| 2E | O_E.Authen_Source | ITENV.1 |
| 3E | O_E.Authen_User | ITENV.6 |
| 4E | O_E.Confidentiality | ITENV.1 |
| 5E | O_E.DAC | ITENV.3 |
| 6E | O_E.Expire | ITENV.3 |
| 7E | O_E.Info_Flow | ITENV.3 |
| 8E | O_E.Integrity | ITENV.1 |
| 9E | O_E.Revoke | ITENV.3<br>ITENV.4 |
| 10E | O_E.Single_Level_Port | ITENV.3 |
| 11E | O_E.SOF | ITENV.1<br>ITENV.2 |
| 12E | O_E.Time | ITENV.7 |
| 13E | O_E.Trusted_Channel | ITENV.1 |
| 14E | O_E.Verify_Config | ITENV.5<br>ITENV.6 |

**Table 8.8 – Mapping of IT Security Objectives for the Environment to Functional Requirements**

### 8.2.2 All Functional Components Necessary

Tables 8.9 and 8.10 show that each functional requirement is necessary, since it is used to address at least one of the IT security objectives.

| | Component | Component Name | Objective |
|---|---|---|---|
| 1 | FAU_GEN.1 | Audit data generation | O.Accountability<br>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<br>O.SOF |
| 11 | FIA_ATD.1 | User attribute definition | O.Authen_Source<br>O.DAC<br>O.Info_Flow |
| 12 | FIA_UAU.2 | User authentication before any action | O.Authen_Source<br>O.Authen_User |
| 13 | FIA_UAU.6 | Re-Authenticating | O.Authen_User |
| 14 | FIA_UID.2 | User identification before any action | O.Authen_Source<br>O.Authen_User |
| 15 | FMT_MOF.1 | Management of Security Functions Behavior | O.Audit_Select<br>O.Revoke |
| 16 | FMT_MTD.1 | Management of TSF Data | O.Audit_Select<br>O.Revoke |
| 17 | FMT_REV.1 | Revocation | O.Revoke |
| 18 | FMT_SAE.1 | Time-limited authorisation | O.Expire |
| 19 | FMT_SMR.1 | Security roles | O.Verify_Config |
| 20 | FPT_ITI.1 | Inter-TSF detection of modification | O.Integrity |
| 21 | FPT_RVM.1 | Non-bypassability of the TSP | O.Non-Bypassability<br>O.Windows_95 |
| 22 | FPT_SEP.1 | TSF domain separation | O.Domain_Separation<br>O.Windows_95 |
| 23 | FPT_STM.1 | Reliable time stamps | O.Time |
| 24 | FPT_TDC.1 | Inter-TSF basic TSF data consistency | O.Consistency |
| 25 | FTP_ITC.1 | Inter-TSF Trusted Channel | O.Trusted_Channel |

**Table 8.9 – Mapping of Functional Requirements to IT Security Objectives**

| Requirement | Requirement for the IT Environment | Objective |
|---|---|---|
| ITENV.1 | Cryptographic Services on Fortezza Card | O_E.Authen_Source<br>O_E.Confidentiality<br>O_E.Integrity<br>O_E.Trusted_Channel<br>O_E.SOF |
| ITENV.2 | Cryptographic Services Strength of Function (SOF) Requirement | O_E.SOF |
| ITENV.3 | Dragonfly Administration System for Setting User Attributes | O_E.Audit_Select<br>O_E.DAC<br>O_E.Expire<br>O_E.Info_Flow<br>O_E.Revoke<br>O_E.Single_Level_Port |
| ITENV.4 | Dragonfly Administration System for Modifying TSF Data | O_E.Audit_Select<br>O_E.Revoke |
| ITENV.5 | Certificates on the Fortezza Card | O_E.Verify_Config |
| ITENV.6 | Fortezza Card PINs | O_E.Authen_User<br>O_E.Verify_Config |
| ITENV.7 | Fortezza Card Time | O_E.Time |

**Table 8.10 – Mapping of IT Environment Requirements to IT Security Objectives**

### 8.2.3 Satisfaction of Dependencies

Table 8.11 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 8.11. 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 | 23 |
| 2 | FAU_SEL.1 | Selective audit | FAU_GEN.1<br>FMT_MTD.1<br>ITENV.3<br>ITENV.4 | 1<br>16<br>-<br>- |
| 3 | FDP_ACC.1 | Subset access control | FDP_ACF.1 | 4 |

| No. | Component | Component Name | Dependencies | Reference |
|---|---|---|---|---|
| 4 | FDP_ACF.1 | Security attribute based access control | FDP_ACC.1<br>FMT_MSA.3<br>ITENV.3 | 3<br>none<br>- |
| 5 | FDP_ETC.1 | Export of user data without security attributes | [FDP_ACC.1 or FDP_IFC.1]<br>ITENV.3 | 3<br>6<br>- |
| 6 | FDP_IFC.1 | Subset information flow control | FDP_IFF.1 | 7 (H) |
| 7 | FDP_IFF.2 | Hierarchical security attributes | FDP_IFC.1<br>FMT_MSA.3<br>ITENV.3 | 6<br>none<br>- |
| 8 | FDP_ITC.1 | Import of user data without security attributes | [FDP_ACC.1 or FDP_IFC.1]<br>FMT_MSA.3<br>ITENV.3 | 3<br>6<br>none<br>- |
| 9 | FDP_UCT.1 | Basic data exchange confidentiality | FTP_ITC.1<br>FDP_IFC.1<br>ITENV.1 | 25<br>6<br>- |
| 10 | FDP_UIT.1 | Data exchange integrity | [FDP_ACC.1 or FDP_IFC.1]<br>FTP_ITC.1<br>ITENV.1 | 3<br>6<br>25<br>- |
| 11 | FIA_ATD.1 | User attribute definition | ITENV.1<br>ITENV.3<br>ITENV.5 | -<br>-<br>- |
| 12 | FIA_UAU.2 | User authentication before any action | FIA_UID.1<br>ITENV.1<br>ITENV.6 | 14 (H)<br>-<br>- |
| 13 | FIA_UAU.6 | Re-Authenticating | None | - |
| 14 | FIA_UID.2 | User identification before any action | None | - |
| 15 | FMT_MOF.1 | Management of Security Functions Behavior | FMT_SMR.1<br>ITENV.3 | 19<br>- |
| 16 | FMT_MTD.1 | Management of TSF Data | FMT_SMR.1<br>ITENV.4 | 19<br>- |
| 17 | FMT_REV.1 | Revocation | FMT_SMR.1<br>ITENV.3<br>ITENV.4 | 19<br>-<br>- |
| 18 | FMT_SAE.1 | Time-limited authorisation | FMT_SMR.1<br>FPT_STM.1<br>ITENV.3 | 19<br>23<br>- |
| 19 | FMT_SMR.1 | Security roles | FIA_UID.1<br>ITENV.5<br>ITENV.6 | 14 (H)<br>-<br>- |
| 20 | FPT_ITI.1 | Inter-TSF detection of modification | ITENV.1 | - |
| 21 | FPT_RVM.1 | Non-bypassability of the TSP | None | - |
| 22 | FPT_SEP.1 | TSF domain separation | None | - |
| 23 | FPT_STM.1 | Reliable time stamps | ITENV.7 | - |
| 24 | FPT_TDC.1 | Inter-TSF basic TSF data consistency | None | - |
| 25 | FTP_ITC.1 | Inter-TSF Trusted Channel | ITENV.1 | - |

**Table 8.11 – 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 Companion. However, the Dragonfly Administration System is used to create the User Fortezza Card for the Dragonfly Companion. The User Fortezza Card contains five certificates: User Certificate, Configuration Certificate, Audit Certificate, Certificate Revocation List, and Routing that contain the security attributes for the Dragonfly Companion. It was deemed acceptable for the Dragonfly Administration System to be outside of the evaluated configuration, even though the Companion depends on it to set its security attributes and update TSF data, because the Dragonfly Companion User interface provides the ability to check the configuration of the Companion.

Because of the way the Dragonfly Companion operates, the Audit Mask and the Certificate Revocation List are both user attributes and TSF data. When a Companion is first initialized, it uses the audit mask and certificate revocation list on its own User Fortezza card. However, when there are multiple Dragonfly Units in a Dragonfly deployment, they periodically exchange audit masks and certificate revocation lists, and each Dragonfly Companion updates itself with the most current values which may come from another Dragonfly Unit. When an audit catcher updates the Audit Mask or Certificate Revocation List of a Companion, 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 Companion on the Dragonfly Administration System. 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 TOE. A Dragonfly Guard serving as an audit catcher also depends upon the Dragonfly Administration System to update its audit mask and certificate revocation list.

### 8.2.5 Auditable Events Rationale

The auditable events provided by the Dragonfly Companion were reviewed against the auditable events for the minimal or basic level of audit for the functional requirements. It was found that the Dragonfly Companion provided auditable events for the applicable functionality in all areas except for confidentiality and integrity. It was decided that it would not be appropriate for the Companion to audit these activities, since all User Data messages sent between a Companion and a Dragonfly Unit have an integrity check applied, and are encrypted for confidentiality. These are routine events for the Dragonfly Companion 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 Strength of Function Rationale

The Strength of Function requirement applies to the following IT Requirement: FDP_UIT.1 - Data exchange integrity, and

A Strength of Function level of SOF-Medium counters the assumed attack level of medium. The strength of function requirement is met by the checksum algorithm for the integrity check (see INT-1).

### 8.2.7 Assurance Requirements Rationale

The Dragonfly Companion claims to satisfy the requirements for EAL2 and no additional assurance requirements. Although the Dragonfly Companion 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 Companion 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 8.12 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 Reference | IT Security Function |
|---|---|---|---|
| FAU_GEN.1 | Audit data generation | AUDIT-1 | Audit Catcher |
| | | AUDIT-2 | Audit Required Configuration Option |
| | | AUDIT-3 | Audit Catcher List |
| | | AUDIT-4 | Audit Catcher Messages |
| | | AUDIT-5 | Audit User Interface |
| | | AUDIT-6 | Auditable Events |
| FAU_SEL.1 | Selective audit | AUDIT-6 | Auditable Events |
| | | AUDIT-7 | Audit Masks |
| | | AUDIT-8 | Audit Mask Management |
| | | AUDIT-9 | Audit Catcher |
| | | AUDIT-10 | Audit Report |
| | | SM-3 | Management of TSF Data |
| FDP_ACC.1 | Subset access control | DAC-1 | Privilege Vectors |
| | | DAC-2 | Shared Domain |
| | | DAC-3 | Block All Mode |
| | | DAC-4 | Intermediate Protection Mode |
| | | DAC-5 | Firewall Protection Mode |
| | | DAC-6 | No Native Associations Routing Option |
| FDP_ACF.1 | Security attribute based access control | DAC-1 | Privilege Vectors |
| | | DAC-2 | Shared Domain |
| | | DAC-3 | Block All Mode |
| | | DAC-4 | Intermediate Protection Mode |
| | | DAC-5 | Firewall Protection Mode |
| | | DAC-6 | No Native Associations Routing Option |
| | | 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 |
| 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 |

| Functional Component | Functional Requirement | TSS Reference | IT Security Function |
|---|---|---|---|
| | | MAC-4 | SMTP Datagrams Blocked for Write Up |
| | | MAC-5 | Allowed Information Flows |
| | | MAC-6 | FTP and SMTP Anticipated Responses |
| | | MAC-7 | Name Server Requests and Responses |
| | | MAC-8 | ICMP Requests and Responses |
| 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 | Name Server Requests and Responses |
| | | MAC-8 | ICMP Requests and Responses |
| | | IP-3 | Dragonfly Ping |
| | | IP-6 | Protected User Datagrams and Security Levels |
| 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 |

| Functional Component | Functional Requirement | TSS Reference | IT Security Function |
|---|---|---|---|
| | | CONF-1 | Confidentiality of User Data |
| FDP_UIT.1 | Data exchange integrity | IP-1 | Types of IP Datagrams |
| | | IP-5 | Encapsulated IP 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 Companion User Fortezza Card |
| | | IA-2 | Fortezza Card Certificate PIN |
| | | IA-3 | Source Authentication |
| FIA_UAU.6 | Re-Authenticating | IA-2 | Fortezza Card Certificate PIN |
| FIA_UID.2 | User identification before any action | IA-1 | Dragonfly Companion User Fortezza Card |
| | | IA-2 | Fortezza Card Certificate PIN |
| | | IA-3 | Source Authentication |
| FMT_MOF.1 | Management of Security Functions Behavior | SM-2 | Dragonfly Administration System |
| | | SM-4 | Configuration Options for Mode |
| | | SM-5 | Allowable Modes |
| | | SM-6 | Mode Set by Trusted Human User |
| | | SM-7 | Reaching Modes |
| FMT_MTD.1 | Management of TSF data | SM-3 | Management of TSF data |
| FMT_REV.1 | Revocation | IA-1 | Dragonfly Companion User Fortezza Card |
| | | IA-2 | Fortezza Card Certificate Pin |
| | | IA-3 | Source Authentication |
| | | ASSOC-2 | Digitally Signed Association Request |
| | | 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_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 IP Datagrams |
| | | IP-5 | Encapsulated IP Datagrams |
| | | INT-2 | Integrity of TSF Data |

| Functional Component | Functional Requirement | TSS Reference | IT Security Function |
|---|---|---|---|
| FPT_RVM.1 | Non-bypassability of the TSP | SA-1 | Non-bypassability of the TSP |
| | | SA-3 | Windows 95 |
| FPT_SEP.1 | TSF domain separation | SA-2 | TSF domain separation |
| | | SA-3 | Windows 95 |
| FPT_STM.1 | Reliable time stamps | TIME-1 | System Time |
| | | TIME-2 | Companion 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 8.12 – Mapping of Functional Requirements to TOE Summary Specification**

### 8.3.2 All TOE Summary Specification (TSS) Functions Necessary

Table 8.13 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 Companion User Fortezza Card | FIA_UID.2 | User identification before any action |
| | | FIA_UAU.2 | User authentication before any action |
| | | FMT_REV.1 | Revocation |
| | | 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 |
| | | FIA_UAU.6 | Re-Authenticating |
| | | FMT_REV.1 | Revocation |
| | | 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 |
| | | FMT_REV.1 | Revocation |
| 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 |
| | | FMT_REV.1 | Revocation |
| | | 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 | Shared Domain | FDP_ACC.1 | Subset access control |
| | | FDP_ACF.1 | Security attribute based access control |
| DAC-3 | Block All Mode | FDP_ACC.1 | Subset access control |
| | | FDP_ACF.1 | Security attribute based access control |
| DAC-4 | Intermediate Protection Mode | FDP_ACC.1 | Subset access control |
| | | FDP_ACF.1 | Security attribute based access control |
| DAC-5 | Firewall Protection Mode | FDP_ACC.1 | Subset access control |
| | | FDP_ACF.1 | Security attribute based access control |
| DAC-6 | No Native Associations Routing Option | 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 |

| TSS Ref No | IT Security Function | Functional Component | Functional Requirement |
|---|---|---|---|
| MAC-7 | Name Server Requests and Responses | FDP_IFC.1 | Subset information flow control |
| | | FDP_IFF.2 | Hierarchical security attributes |
| MAC-8 | ICMP Requests and Responses | 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 IP Datagrams | FPT_ITI.1 | Inter-TSF detection of modification |
| IP-5 | Encapsulated IP 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 Catcher | 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 User Interface | 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 |
|---|---|---|---|
| AUDIT-9 | Audit Catcher | FAU_SEL.1 | Selective Audit |
| AUDIT-10 | Audit Report | FAU_SEL.1 | Selective Audit |
| 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 |
| TIME-2 | Companion 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 |
| | | FMT_MOF.1 | Management of Security Functions Behavior |
| SM-3 | Management of TSF Data | FAU_SEL.1 | Selective Audit |
| | | FMT_MTD.1 | Management of TSF Data |
| | | FMT_REV.1 | Revocation |
| SM-4 | Configuration Options for Mode | FMT_MOF.1 | Management of Security Functions Behavior |
| SM-5 | Allowable Modes | FMT_MOF.1 | Management of Security Functions Behavior |
| SM-6 | Mode Set by Trusted Human User | FMT_MOF.1 | Management of Security Functions Behavior |
| SM-7 | Reaching Modes | FMT_MOF.1 | Management of Security Functions Behavior |
| 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 |
| SA-3 | Windows 95 | FPT_RVM,1 | Non-bypassability of the TSP |
| | | FPT_SEP.1 | TSF Domain Separation |

**Table 8.13 – Mapping of TOE Summary Specification to Functional Requirements**

### 8.3.3 Assurance Measures Rationale

Table 8.14 shows that all of the EAL2 Assurance requirements are satisfied.

| Component | Component Title | Evidence Requirements | How Sati |
|---|---|---|---|
| ACM_CAP.2 | Configuration items | CM Documentation | Companic Configura |
| ADO_DEL.1 | Delivery procedures | Delivery Procedures | Dragonfly Shipping I<br>Guard De<br>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 Informal F Specificat |
| ADV_HLD.1 | Descriptive high-level design | High-Level Design | Dragonfly Descriptiv Design Dc |
| ADV_RCR.1 | Informal correspondence demonstration | Representation Correspondence | Dragonfly Informal 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 Informal Demonstr |
| ATE_FUN.1 | Functional testing | Test Documentation | Test Plans Test Resu |
| ATE_IND.2 | Independent testing – sample | TOE for Testing | TOE for T<br>Evaluation (See Sect Testing of Report.) |
| AVA_SOF.1 | Strength of TOE security function evaluation | SOF Analysis | 99-003, D Checksun |
| AVA_VLA.1 | Developer vulnerability analysis | Vulnerability Analysis | Vulnerabil Dragonfly |

**Table 8.14 – 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** | Multilevel 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 Companion**


| | |
|---|---|
| DF_AUM | ITT Industries, Dragonfly Administration User Manual, Version 2.05c, October 19, 1999. |
| DF_CD | ITT Industries, Dragonfly Companion Informal Correspondence Demonstration, Version 1.02, 25 October 1999; |
| DF_CM | ITT Industries, S. Meloche Memo 99-007e, October 29, 1999; Subject: Companion TOE Configuration Management; |
| DF_HLD | ITT Industries, *Dragonfly Companion Descriptive High Level Design Document,* Version 1.8, 28 June 1999; |
| DF_IFS | ITT Industries, *Dragonfly Companion Informal Functional Specification,* Version 1.1, 19 May 1999; |
| DF_IMSTMT | ITT Industries, Dr. E. Wrench , Impact Statement for Dragonfly Companion TOE Security Functions Change, 19 May 1999; |
| DF_TPROC | ITT Industries, *Dragonfly Test Procedures*, Version 3.01a, 20 May 1999; |
| DF_UM | ITT Industries, *Dragonfly Companion User Manual*, Version 2.05d, October 25, 1999; |
| DF_VA | ITT Industries, *Vulnerability Analysis of the Dragonfly Companion,* Version 1.1, 23 June 1999; |
| 99-003 | ITT Industries, S. Levin Memo 99-003; March 22, 1999, Subject: Dragonfly Companion 32-bit Checksum; |
| 99-004 | ITT Industries, S. Levin Memo 99-004, March 22, 1999; Subject: Dragonfly Anticipated Messages; |
| 99-023 | ITT Industries, S. Levin Memo 99-023b, October 12, 1999; Subject: Dragonfly Companion Shipping Procedures; |
| 99-024 | ITT Industries, S. Levin Memo 99-024, September 21, 1999; Subject: Guard Delivery Procedures; |


**Dragonfly Guard**


| | |
|---|---|
| DF_GCM | S. Levin Memo 98-016f; October 22, 1998 Subject: Guard TOE Configuration Management; |
| DF_GFER | *ITT Industries Dragonfly Guard Final Evaluation Report,* Version 1.1, 29 October 1998. |
| DF_GST | *ITT Industries Dragonfly Guard Security Target*, Version 2.0, 29 October 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.