DEPARTMENT OF DEFENSE

MAIL GUARD FOR HIGH ROBUSTNESS ENVIRONMENTS

PROTECTION PROFILE

VERSION 0.1

September 30, 2001

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	9/30/2001	Report 9/30/2001

4. TITLE AND SUBTITLE
Department of Defense Mail Guard for High Robustness Environments

5. FUNDING NUMBERS

6. AUTHOR(S)
Gilmore, Linda M,; Mayer, Barbara; Montequin, Rita; Rogers, Kristina; Weiss, Howard

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Booz Allen & Hamilton
8283 Greensboro Drive
McLean, VA 22102

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of Defense

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; Distribution unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT ***(Maximum 200 Words)***

This Protection Profile (PP) specifies the information security requirements for DoD Mail Guards for High Robustness Environments. The Mail Guard specified in this PP sits between two protected network enclaves at different classification levels, controlling the flow of electronic messages sent between the two networks. The protection approach employs various processing, filtering, and data-blocking techniques in an attempt to provide data sanitization (e.g., downgrade) or separation between enclaves. Besides enforcing an information flow policy and providing services for confidentiality and integrity of mail messages, the Mail Guard provides identification and authentication, trusted path and audit capabilities and has been designed with a high degree of assurance. The specific functional and assurance requirements are contained in Section 5 of this document

14. SUBJECT TERMS
IATAC Collection, information security, protection profile, mail guard

15. NUMBER OF PAGES
76

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

Protection Profile Title:

Department of Defense (DoD) Mail Guard for High Robustness Environments Protection Profile (PP).

Criteria Version:

This Protection Profile (PP) was developed using Version 2.1 of the Common Criteria (CC) [1].

Constraints:

Targets of Evaluation (TOEs) developed to satisfy this Protection Profile shall conform to CC Part 2 and CC Part 3.

Authors:

This Protection Profile was prepared by:

Linda M. Gilmore, SPARTA, Inc.

Barbara Mayer, SPARTA, Inc.

Rita Montequin, SPARTA, Inc.

Kristina Rogers, CygnaCom Solutions, Inc.

Howard Weiss, SPARTA, Inc.

Acknowledgements:

The authors would like to acknowledge Kenneth W. Eggers from CygnaCom Solutions, Inc.

TABLE OF CONTENTS

1.0 PROTECTION PROFILE INTRODUCTION 1
1.1 Protection Profile Identification 1
1.2 Protection Profile Overview 1
1.3 Conventions 1
1.4 Terminology 2
1.5 PP Organization 3
3.0 SECURITY ENVIRONMENT 7
3.1 Secure Usage Assumptions 7
3.2 Organizational Security Policies 8
3.3 Threats Addressed by the TOE 8
3.4 Threats to the Environment 10
4.0 SECURITY OBJECTIVES 11
4.1 TOE Security Objectives 11
4.2 Security Objectives for the Environment 13
5.0 SECURITY REQUIREMENTS 15
5.1 TOE Security Functional Requirements 15
5.1.1 Security Audit (FAU) 16
5.1.2 Cryptographic Support (FCS) 21
5.1.3 User Data Protection (FDP) 21
5.1.4 Identification and Authentication (FIA) 26
5.1.5 Security Management (FMT) 27
5.1.6 Protection of the TOE Security Functions (FPT) 30
5.1.7 Trusted Path (FTP) 32
5.2 Security Requirements for the Environment 33
5.3 TOE Security Assurance Requirements 34
5.3.1 Configuration Management (ACM) 35
5.3.2 Delivery and Operation (ADO) 37
5.3.3 Development (ADV) 38
5.3.4 Guidance Documents (AGD) 43
5.3.5 Life Cycle Support (ALC) 45
5.3.6 Testing (ATE) 47
5.3.7 Vulnerability Assessment (AVA) 49
6.0 RATIONALE 53
6.1 Rationale for TOE Security Objectives 53
6.2 Rationale for Security Objectives/Requirements for the Environment 58
6.3 Rationale for Security Requirements 58
6.4 Rationale for Assurance Requirements 66
6.5 Rationale for Not Satisfying All Dependencies 67
6.6 Rationale for Strength of Function Claim 67
Appendix A: Acronyms 68

Appendix B: References 69

Appendix C: EAL Table 70

LIST OF FIGURES AND TABLES

Figure 1 - Mail Guard Architecture 4

Table 1 - Security Functional Requirements 16

Table 2 - Auditable Events 19

Table 3 - Management of Security Attributes 29

Table 4 - Management of TOE Data 30

Table 5 - Security Objectives to Threats/Policies Mapping 57

Table 6 - Functional Requirements to Security Objectives Mapping 65

1.0 PROTECTION PROFILE INTRODUCTION

1.1 PROTECTION PROFILE IDENTIFICATION

1 **Title:** Department of Defense (DoD) Mail Guard for High Robustness Environments Protection Profile

2 **Sponsor:** National Security Agency (NSA)

3 **Authors:** Linda M. Gilmore, Barbara Mayer, Rita Montequin, Kristina Rogers, and Howard Weiss

4 **Contributor:** Kenneth W. Eggers

5 **CC Version:** Common Criteria (CC) Version 2.1

6 **Registration:** <to be provided upon registration>

7 **PP Version:** Version 0.1, dated 30 September 2001

8 **Keywords:** Guard, Mail Guard, Mail Transfer Agent, Simple Mail Transfer Protocol

1.2 PROTECTION PROFILE OVERVIEW

9 This Protection Profile (PP) specifies the information security requirements for DoD Mail Guards for High Robustness Environments. The Mail Guard specified in this PP sits between two protected network enclaves at different classification levels, controlling the flow of electronic messages sent between the two networks. The protection approach employs various processing, filtering, and data-blocking techniques in an attempt to provide data sanitization (e.g., downgrade) or separation between enclaves. Besides enforcing an information flow policy and providing services for confidentiality and integrity of mail messages, the Mail Guard provides identification and authentication, trusted path and audit capabilities and has been designed with a high degree of assurance. The specific functional and assurance requirements are contained in Section 5 of this document.

1.3 CONVENTIONS

10 The notation, formatting, and conventions used in this Protection Profile are largely consistent with those used in version 2.1 of the Common Criteria (CC). Selected presentation choices are discussed here to aid the PP user.

11 The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in paragraph 2.1.4 of Part 2 of the CC. Each of these operations is used in this PP.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by **bold text**.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections are denoted by *underlined italicized text*.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignment is indicated by showing the value in square brackets, [assignment_value].

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing the iteration number in parenthesis following the component identifier, (iteration_number).

The **security target writer** operation is used to denote points in which the final determination of attributes is left to the security target writer. Security Target writer operations are indicated by the words {to be determined by the Security Target writer} in braces.

12 Application Notes are provided to help the developer, either to clarify the intent of a requirement, identify implementation choices, or to define "pass-fail" criteria for a requirement. For those components where Application Notes are appropriate, the Application Notes will follow the requirement component.

1.4 TERMINOLOGY

13 In the CC, many terms are defined in Section 2.3 of Part 1. The following definitions are listed here to aid the user's understanding of this PP.

Authorized Administrator - A role which human users may be associated with to administer the security parameters of the TOE. An Authorized Administrator is not subject to any access control requirements once authenticated to the TOE and is therefore trusted to not compromise the security policy enforced by the TOE. The Authorized Administrator is responsible for administering the TOE (i.e., operating system configuration) security parameters.

Directory – One or more independently operated and distributed Directory Service Agents (DSAs) that provide information to support White Pages users (e.g., name, address and telephone number) and PKI users (e.g., email address, public key certificates and revocation information).

Directory Service Agent – As defined in RFC 1943, an application that offers the directory service, that is, the database for the Directory.

Directory User Agent (DUA) – As defined in RFC 1943, an application that facilitates user access to a DSA.

Guard Application Administrator – A role which human users may be associated with to administer the Mail Guard Application. The Guard Application Administrator is responsible for administering the security parameters for the guard application (i.e., filter settings).

Mail Transfer Agent (MTA) – A software program responsible for the delivery of electronic mail. The MTA receives mail from a User Agent (UA) or another MTA, and then performs the routing and delivery functions.

Mail Transfer System (MTS) – A collection of MTAs that transfers messages from an originating UA to a recipient UA.

User Agent (UA) – A process that makes the services of the MTS available to the user. A UA may be implemented as a computer program that provides utilities to create, send, receive, and perhaps archive messages.

1.5 PP ORGANIZATION

14 Section 1, PP Introduction, provides document management and overview information necessary to identify the PP along with references to other related PPs.

15 Section 2, Target of Evaluation (TOE) Description, defines the TOE and establishes the context of the TOE by referencing generalized security requirements.

16 Section 3, TOE Security Environment (TSE), describes the expected environment in which the TOE is to be used. This section defines the set of threats that are relevant to the secure operation of the TOE, organizational security policies with which the TOE must comply, and secure usage assumptions applicable to this analysis.

17 Section 4, Security Objectives, defines the set of security objectives to be satisfied by the TOE and by the TOE operating environment.

18 Section 5, IT Security Requirements, defines the functional and assurance requirements derived from the Common Criteria, Part 2 and Part 3, respectively, that must be satisfied by the TOE.

19 Section 6, Rationale, provides rationale to demonstrate that the security objectives satisfy the threats and polices. This section also explains how the set of requirements are complete relative to the security objectives and presents a set of arguments that address dependency analysis and Strength of Function (SOF).

20 Expansion of acronyms are provided to facilitate comprehension of frequently used terms.

21 References are provided as background material for further investigation by interested users of the Protection Profile.

2.0 TOE DESCRIPTION

22 The DoD Mail Guard for High Robustness Environments, hereafter referred to as the Target Of Evaluation (TOE), has the ability to separate networks of different classifications, assuring that the only electronic mail traffic allowed to pass between networks is that allowed by a site-defined security policy. As shown in Figure 1, the TOE stands between two classified network enclaves where the security levels (i.e., CLASSIFICATION A and CLASSIFICATION B) are not the same. The TOE supports both ISO X.400 and IETF electronic messaging standards, as well as ISO X.500 and IETF LDAP directory service standards.[1]

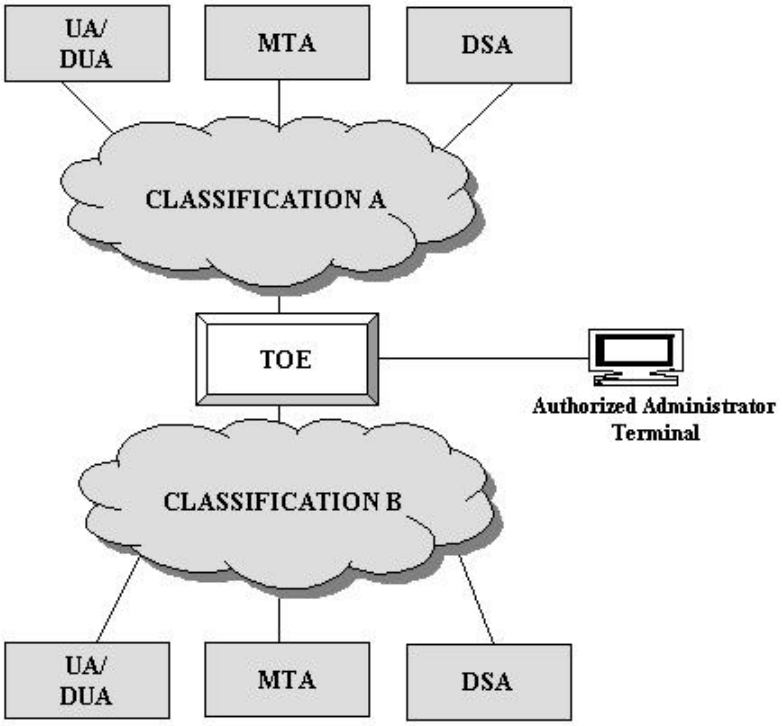

Figure 1 - Mail Guard Architecture

23 The TOE is configurable to support various message release policies from a protected enclave and admittance of messages into a protected enclave. Specifically, the TOE allows each enclave to control the flow of X.400 and RFC 822 messages, and X.500 directory data transfers into and out of the enclave in accordance with a set of release and admittance policies.[2] Filter options include host, sender and receiver access control lists;

---

[1] The IETF electronic messaging standards define the Simple Mail Transfer Protocol (SMTP), as specified in RFC 821, the Internet Message Format, as specified in RFC 822, and Multipurpose Internet Mail Extensions (MIME), as specified in RFC 1341.

[2] The TOE does not perform cross-transfer between X.400 and SMTP messages. That is, the TOE does not do protocol conversion.

use of encryption; security level; allowable attachment types; and plain text string search. The TOE supports enforcement of the DoD Mandatory Access Control policy by limiting flows at different security levels to those that are consistent with the overall system policy.

24 Typical flows for messages and directory information are as follows. When an inbound Mail Transfer Agent (MTA) receives a message, the message is forwarded, according to the messaging protocol used and the outbound MTA for which it is intended, to a set of X.400 or RFC 822 security policy filters appropriate for that data flow. The filters check all relevant message characteristics – envelope, header, encryption, details, content, and disposition – against the associated rules in the site-defined X.400 or RFC 822 security policy that was configured in the TOE for that particular data flow. If validated by the filter set, the message is then reclassified and sent to an outbound MTA at the level of the intended recipient. The outbound MTA then routes the message to the destination User Agent (UA) or to the next MTA in the routing chain. If not validated by the filter set, an audit record is generated and the TOE deletes the message.

25 When an inbound Directory Service Agent (DSA) receives a request over the TOE-connected network at the same security level, it first authenticates itself and the requesting Directory User Agent (DUA) or DSA. Strong authentication using X.509 Version 3 certificates must be used. Once the request is authenticated, the inbound DSA passes the request through its X.500 or LDAP security policy filters. The security policy filters ensure that the message conforms to the release policy configured for the directory data flow between the requesting DUA or DSA and the responding DSA. If the request passes the security filter checks, the inbound DSA reclassifies the message and sends it to the filtering application's DSA at the new security level (i.e., the level of the destination network enclave). The outbound DSA establishes a connection between itself and the destination DSA, authenticating that connection using strong authentication. It then sends the message to the destination DSA.

26 In addition to supporting a flow control policy, the TOE provides user identification and authentication, trusted path to and from the cryptographic module, trusted facility management (i.e., separate Guard Application and Authorized Administrator functions), and trusted recovery. Authorized Administrators and Guard Application Administrators must authenticate themselves to the TOE. Technologies used by the TOE to authenticate the Authorized Administrator and Guard Application Administrator to the TOE include, but are not limited to, one-time passwords, digital certificates or biometrics. Authorized Administrators and Guard Application Administrators must administer the TOE locally via a physically protected direct connection to a console port. The Authorized Administrator is responsible for administering the TOE (i.e., operating system configuration) security parameters. The Guard Application Administrator is responsible for administering the security parameters of the guard application (i.e., filter settings).

27 The TOE is capable of auditing all message traffic; use of identification and authentication mechanisms; actions taken by the Authorized Administrator and/or Guard Application Administrator; changes made to the TOE's security policy rules and data; and changes made to the TOE's date and time; and the use of other security functions.

The decision to record auditable events will be made in accordance with the organizational security policy and will be implemented by the Authorized Administrator. The TOE will be able to include or exclude auditable events recorded based on a set of attributes (at a minimum IP address, type of service (SMTP, X.400, X.500 or LDAP), security level, named sender, named recipient, type of attachment, and encrypted/unencrypted messages). Audit trail data is stamped with a dependable date and time when recorded. If the audit trail becomes full then the only auditable events that are recorded are those performed by the Authorized Administrator. The TOE will take action to notify the Authorized Administrator when the audit trail exceeds 90% capacity.

28 TOEs meeting this PP shall verify digital signatures according to the Digital Signature Algorithm (as specified in FIPS PUB 186-2), perform encryption/decryption using an NSA-certified high robustness algorithm, and compute a secure hash using the Secure Hash Algorithm (SHA-1) (as specified in FIPS PUB 180-1).

29 The TOE assurance requirements are selected to provide a high degree of confidence that the Mail Guard functions as designed, are tamper-proof and non-bypassable. The assurance requirements specified in this PP are greater than those required for Evaluation Assurance Level 4 (EAL4). Specifically, the TOE must satisfy the configuration management and delivery and operation assurance requirements at EAL4 to ensure that modifications during delivery are detected and tracking of changes and security flaws are reported. The TOE must meet the development assurance requirements up to EAL6 to ensure that the high-level and low-level design are described in a semi-formal manner and are supported by a semi-formal security policy model. The Guidance Documents and Life Cycle Support requirements must satisfy at least the EAL4 requirements; additionally, the TOE must be developed by a controlled process. The Testing and Vulnerability Assessment must meet the EAL6 requirements to ensure that the functional testing, covert channel analysis, and a thorough analysis for vulnerabilities are performed. The specific assurance requirements for the TOE are documented in Section 5.

3.0 SECURITY ENVIRONMENT

30 The *Guidance and Policy for the Department of Defense Global Information Grid Information Assurance (GIG)* [3] requires that all information systems employ protection mechanisms according to the level of robustness required relative to the sensitivity of the data to be protected and the threat agents likely to be involved. TOEs compliant with this PP are intended to be used in a High Robustness Environment (HRE). High Robustness is defined in the GIG policy as: "security services and mechanisms that provide thorough rigorous analysis, the most confidence in the security countermeasures". High robustness technical solutions are required by the GIG to include all of the following:

- NSA-certified high-robustness cryptography (algorithms and implementation) for encryption, key exchange, digital signature, and hash;
- NSA-certified high-robustness cryptographically authenticated access control (e.g., digital signature, public key cryptography based, challenge/response identification and authentication);
- NSA-approved key management for symmetric key;
- Class 5 PKI certificates for asymmetric key; and
- High Assurance security design that meets at a minimum Evaluated Assurance Level (EAL) 4, as defined in the Common Criteria (CC).

31 The remainder of this section addresses assumptions about the security aspects of a compliant TOE environment, threats to TOE assets or to the TOE environment that must be countered, and organizational security policies that compliant TOEs must enforce.

3.1 SECURE USAGE ASSUMPTIONS

A.CRYPTOGRAPHY

32 The cryptographic algorithm and key lengths are assumed to be strong enough to counter a high level of attack.

A.NO_EVIL_PROGRAMS

33 There are no untrusted user programs on the TOE.

A. NO_EVIL_USERS

34 Authorized Administrators and Guard Application Administrators are non-hostile, appropriately trained and follow all administrator guidance. However, they are capable of error.

A.PHYSICAL_SECURITY

35 The TOE will reside in a physically secure environment.

A.TOE_ENTRY_POINT

36 Information cannot flow between the two enclaves without passing through the TOE.

3.2 ORGANIZATIONAL SECURITY POLICIES

P.CRYPTOGRAPHY

37 The TOE shall utilize cryptographic modules that are compliant with the GIG.

P.MANDATORY_ACCESS_CONTROL

38 A mandatory access control policy based on hierarchical security levels and categories shall be enforced. Information shall not be allowed to flow from a higher security level to a lower security level or between non-comparable security levels.

3.3 THREATS ADDRESSED BY THE TOE

T.ADDRESS_SPOOFING

39 A threat agent may circumvent the TOE's security policy by spoofing the source address in order to masquerade as another user.

T.ADMINISTRATION

40 A threat agent may make an error in the management of the TOE. Also, a threat agent may cause an error due to being given more privileges than required.

T.AUDIT_FULL

41 A threat agent may cause audit records to be lost or prevent future records from being recorded by taking actions to exhaust storage capacity, thus masking an attacker's actions.

T.AUDIT_UNDETECTED

42 A threat agent may cause auditable events to go undetected.

T.BRUTE_FORCE

43 A threat agent may repeatedly try to guess authentication data in order to launch an attack against the TOE.

T.BYPASS

44 A threat agent may attempt to bypass the security enforcing functions of the TOE.

T.COVERT_CHANNEL

45 A threat agent may use an entity not normally viewed as a data container (e.g., object) to transfer information from a container at one security level to a container at another security level.

T.CRYPTOGRAPHIC_ATTACK

46 A threat agent, using a cryptographic attack, may obtain information for which they are not authorized.

T.DISCLOSURE

47 A threat agent may be able to gain access to information that is released in violation of the TOE security policy due to lack of confidentiality protection.

T.EXCESS_AUDIT

48 A threat agent may cause an Authorized Administrator to be unable to analyze audit data due to an excess volume of data being recorded.

T.HIGH_ATTACK_POTENTIAL

49 A threat agent possessing high attack potential may attempt to bypass or tamper with the TOE security functions to gain access to the TOE or the assets it protects.

T.IDENTIFICATION_AUTHENTICATION

50 A threat agent may attempt to perform actions on the TOE without being held accountable for their actions.

T.INCORRECT_LEVEL

51 A threat agent may cause information at a higher security level to be released to an enclave at a lower security level.

T.MASQUERADE

52 A threat agent may masquerade as the TOE thereby capturing valid identification and authentication data.

T.MODIFY_DATA

53 A threat agent may attempt to modify or destroy security-critical TOE data or programs.

T.REPLAY

54 A threat agent may capture and replay valid identification and authentication information to disguise itself as an Authorized Administrator or Guard Application Administrator of the TOE.

T.SECURITY_LEVEL

55 A threat agent may cause data to be improperly protected due to the TOE's inability to correctly associate a security level with the data on export or import.

T.SYSTEM_FAILURE

56 A threat agent may cause the TOE to perform incorrectly resulting in a system failure.

3.4 THREATS TO THE ENVIRONMENT

T.KEY_COMPROMISE

57 A threat agent, through the use of stolen or compromised cryptographic keys, may decrypt and gain unauthorized access to sensitive data.

4.0 SECURITY OBJECTIVES

4.1 TOE SECURITY OBJECTIVES

O.ACCOUNTABILITY

58 The TOE must be able to hold all users accountable for their actions. It must be possible to identify the user responsible for performing an action or sending a message. Security relevant events must be associated with the identity of the user. It must be possible to verify the sender of a message.

O.ADMIN_SUPPORT

59 The TOE must provide administrative tools to enable Authorized Administrators and Guard Application Administrators to effectively manage and maintain the TOE. The TOE must support these administrators in the performance of their duties and be designed to reduce the likelihood of administrative errors. The TOE must require a user to take an action before assuming an administrator role.

O.AUDIT

60 The TOE must provide a means to accurately detect and record security-relevant events in audit records. The TOE must detect and notify the Authorized Administrator and/or the Guard Application Administrator when the audit log becomes full.

O.AUDIT_PROTECT

61 The TOE must protect the audit log from deletion and modification.

O.AUDIT_SELECT

62 The TOE must be able to change the selection of auditable events during normal operation.

O.AUTHENTICATION

63 The TOE must require that Authorized Administrators and Guard Application Administrators be authenticated (via a single-use authentication mechanism) before performing any TSF-mediated activities. Authentication of information passing through the TOE must be based on cryptographic mechanisms. The TOE must prevent brute force attacks by limiting the number of authentication attempts allowed in a session.

O.CONFIDENTIALITY

64 The TOE must be able to protect messages and other data from unauthorized disclosure.

O.COVERT_CHANNEL

65 The TOE must limit the number (i.e., capacity) and type of illicit information flows between security levels.

O.CRYPTOGRAPHY

66 The cryptographic module used in the TOE must be compliant with the GIG.

O.DATA_INTEGRITY

67 The TOE must be able to verify that messages and other data have not been modified.

O.DOMAIN_SEPARATION

68 The TOE must maintain its own domain for execution and ensure that it cannot be interfered with or tampered with by an untrusted subject.

O.IMPERSONATE

69 The TOE must provide a trusted path for Authorized Administrators and Guard Application Administrators to assure that they are communicating with the TOE when entering authentication information.

O.INFORMATION_FLOW

70 The TOE must not release information from a higher-level enclave to a lower level enclave or between non-comparable levels.

O.MULTI_LEVEL_PORT

71 The TOE must ensure that messages with trusted security labels are interpreted correctly on import and export from/to the TOE.

O.NON-BYPASSABILITY

72 The TOE must ensure that a message cannot be released unless the configured filters are invoked and succeed.

O.RECOVERY

73 The TOE must be able to recover to a secure state in the event of system failure.

O. ROLE_SEPARATION

74 The TOE must provide separate roles for the Authorized Administrator and the Guard Application Administrator.

O.SELF_PROTECT

75 From its initial startup, the TOE must protect itself against attempts to modify, deactivate, or circumvent the TOE security functions.

O.SELF_TEST

76 A TOE must provide and execute self-tests during initial start-up, at the request of the security administrator, and during automated recovery to verify the integrity of its code and data structures.

O.SINGLE_LEVEL_PORT

77 For messages entering/exiting the TOE, the TOE must attach a label to all unlabeled data equal to the level of the source enclave.

O.SOF

78 The TOE must be able to meet strength of function equivalent to SOF-high.

4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT

OE.CRYPTOGRAPHY

79 The cryptographic algorithms and key lengths are assumed to be strong enough to counter a high level of attack.

OE.NO_EVIL_PROGRAMS

80 There are no untrusted user programs on the TOE.

OE. NO_EVIL_USERS

81 Authorized Administrators and Guard Application Administrators are non-hostile, appropriately trained and follow all administrator guidance. However, they are capable of error.

OE.KEY_PROTECTION

82 The TOE must protect the confidentiality and integrity of cryptographic keys during key generation, key distribution and key destruction.

OE.PHYSICAL_SECURITY

83 The TOE will reside in a physically secure environment.

OE.TOE_ENTRY_POINT

84 Mail cannot flow between the two enclaves without passing through the TOE.

5.0 SECURITY REQUIREMENTS

85 This section provides functional and assurance requirements that must be satisfied by a Protection Profile-compliant TOE. These requirements consist of functional components from Part 2 of the CC and assurance components from Part 3 of the CC.

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS

86 The security functional requirements for the TOE are summarized in Table 1 below[3]. The functional components are presented in alphabetical order by component name in the CC.

Functional Components	
FAU_GEN.1	Audit data generation
FAU_SAA.1	Potential violation analysis
FAU_SEL.1	Selective audit
FAU_STG.1	Protected audit trail storage
FAU_STG.3	Action in case of possible audit data loss
FAU_STG.4	Prevention of audit data loss
FCS_COP.1	Cryptographic operation
FDP_ETC.1	Export of user data without security attributes
FDP_ETC.2	Export of user data with security attributes
FDP_IFC.1	Subset information flow control
FDP_IFF.2	Hierarchical security attributes
FDP_IFF.3	Limited illicit information flows
FDP_ITC.1	Import of user data without security attributes
FDP_ITC.2	Import of user data with security attributes
FDP_RIP.2	Full residual information protection
FIA_AFL.1	Authentication failure handling
FIA_ATD.1	User attribute definition
FIA_UAU.2	User authentication before any action
FIA_UAU.4	Single-use authentication mechanisms
FIA_UID.2	User identification before any action

[3] Iterations of the same component are not repeated in the table.

Functional Components	
FMT_MOF.1	Management of security functions behaviour
FMT_MSA.1	Management of security attributes
FMT_MSA.2	Secure security attributes
FMT_MSA.3	Static attribute initialisation
FMT_MTD.1	Management of TSF data
FMT_SMR.2	Restrictions on security roles
FMT_SMR.3	Assuming roles
FPT_AMT.1	Abstract machine testing
FPT_ITT.1	Basic internal TSF data transfer protection
FPT_RCV.2	Automated recovery
FPT_RPL.1	Replay detection
FPT_RVM.1	Non-bypassability of the TSP
FPT_SEP.2	SFP domain separation
FPT_STM.1	Reliable time stamps
FPT_TDC.1	Inter-TSF basic TSF data consistency
FPT_TST.1	TSF testing
FTP_ITC.1	Inter-TSF trusted channel
FTP_TRP.1	Trusted path

Table 1 - Security Functional Requirements

5.1.1 SECURITY AUDIT (FAU)

FAU_GEN.1 Audit data generation

87 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 *basic* level of audit; and

c) [the events in Table 2].

Functional Component	Auditable Event	Additional Audit Record Contents
FAU_SAA.1	Enabling and disabling of any of the analysis mechanisms.	The identity of the Authorized Administrator performing the operation.
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.	The identity of the Authorized Administrator performing the change to the audit configuration and the audit parameter changed.
FAU_STG.3	Actions taken due to exceeding the threshold.	The identity of the Authorized Administrator performing the operation.
FAU_STG.4	Actions taken due to audit storage failure.	The identity of the Authorized Administrator performing the operation.
FCS_COP.1	Success and failure and the type of cryptographic operation. Any applicable cryptographic mode(s) of operation, subject attributes and object attributes.	The presumed addresses of the source and destination.
FDP_ETC.1	All attempts to export information.	The presumed addresses of the source and destination.
FDP_ETC.2	All attempts to export information.	The presumed addresses of the source and destination.
FDP_IFF.2	All decisions on requests for information flow.	The presumed addresses of the source and destination.
FDP_IFF.3	All decisions on requests for information flow and the use of identified illicit information flow channels.	The presumed addresses of the source and destination.
FDP_ITC.1	All attempts to import user data including any security attributes.	The presumed addresses of the source and destination.
FDP_ITC.2	All attempts to import user data including any security attributes.	The presumed addresses of the source and destination.
FIA_AFL.1	Reaching the threshold of unsuccessful authentication attempts and the actions taken and the subsequent, and restoration to normal operational state.	The identity of the offending user and the Authorized Administrator.

Functional Component	Auditable Event	Additional Audit Record Contents
FIA_UAU.2	All use of the authentication mechanism.	The user identities presented to the TOE.
FIA_UAU.4	Attempts to reuse authentication data.	The user identities presented to the TOE.
FIA_UID.2	All use of the user identification mechanism, including the user identity.	The user identities presented to the TOE.
FMT_MOF.1	All modifications in the behavior of the functions in the TOE.	The identity of the Authorized Administrator and/or Guard Application Administrator performing the modification.
FMT_MSA.1	All modifications of the values of security attributes.	The identity of the Authorized Administrator and/or Guard Application Administrator performing the modification.
FMT_MSA.2	All offered and rejected values for a security attribute.	The identity of the Authorized Administrator and/or Guard Application Administrator performing the modification.
FMT_MSA.3	Modifications of the default setting of permissive or restrictive rules and all modifications of the initial values of the security attributes.	The identity of the Authorized Administrator and/or Guard Application Administrator performing the modification.
FMT_MTD.1	All modifications to the limits on TOE data.	The identity of the Authorized Administrator and/or Guard Application Administrator performing the modification.
FMT_SMR.2	Modifications to the group of users that are part of a role and unsuccessful attempts to use a role due based on the conditions of the role.	The identity of the Authorized Administrator and/or Guard Application Administrator performing the modification.
FMT_SMR.3	Explicit request to assume a role.	The identity of the Authorized Administrator and/or Guard Application Administrator performing the operation.
FPT_AMT.1	Execution of the tests of the underlying machine and the results of the tests.	The identity of the Authorized Administrator performing the operation.

Functional Component	Auditable Event	Additional Audit Record Contents
FPT_RCV.2	The fact that a failure of service discontinuity occurred and the resumption of the regular operation. Type of failure or service discontinuity.	Actions taken to recover the TOE to a secure state.
FPT_RPL.1	Detected replay attacks.	The presumed addresses of the source and destination.
FPT_STM.1	Changes to the time.	The identity of the Authorized Administrator performing the operation.
FPT_TDC.1	Use of TOE data consistency mechanisms, identification of which TOE data have been interruption, and detection of the modified TOE data.	The presumed addresses of the source and destination.
FPT_TST.1	Execution of the TOE self tests and the results of the tests.	The identity of the Authorized Administrator performing the operation.
FTP_ITC.1	All attempted uses of the trusted channel functions and identification of the initiator and target of all trusted channel functions.	The identity of the Authorized Administrator and/or Guard Application Administrator performing the operation.
FTP_TRP.1	All attempted uses of the trusted path functions and identification of the user associated with all trusted path invocations.	The user identities presented to the TOE.

Table 2 - Auditable Events

88 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, *source IP address, destination IP address, service (SMTP, LDAP, X.500 or X.400), packet data, other data {to be determined by the Security Target writer}*.

FAU_SAA.1 Potential violation analysis

89 FAU_SAA.1.1 - The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP. FAU_SAA.1.2 - The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation or combination of [unsuccessful use of authentication mechanisms] known to indicate a potential security violation; and

b) [other events {to be determined by the Security Target writer}].

FAU_SEL.1 Selective audit

90 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 attributes:

a) *event type*; and

b) [IP address, service (SMTP, LDAP, X.500 or X.400), security level, named sender, named recipient, type of attachment, encrypted/unencrypted messages and other attributes {to be determined by the Security Target writer}].

FAU_STG.1 Protected audit trail storage

91 FAU_STG.1.1 - The TSF shall protect the stored audit records from unauthorised deletion.

92 FAU_STG.1.2 - The TSF shall be able to *prevent* modifications to the audit records.

FAU_STG.3 Action in case of possible audit data loss

93 FAU_STG.3.1 - The TSF shall take [measures to notify the Authorized Administrator] if the audit trail exceeds [90% storage capacity].

FAU_STG.4 Prevention of audit data loss

94 FAU_STG.4.1 - The TSF shall take *prevent auditable events, except those taken by the Authorized* ***Administrator*** and [shall limit the number of audit records lost] if the audit trail is full.

95 Application Note: The Security Target writer is expected to provide, as part of the "Security Requirements Rationale" section, an analysis of the maximum amount of audit data that can be expected to be lost in the event of audit storage failure, exhaustion, and/or attack.

5.1.2 CRYPTOGRAPHIC SUPPORT (FCS)

FCS_COP.1 (1) Cryptographic operation

96 FCS_COP.1.1 - The TSF shall [verify digital signatures] in accordance with a specified cryptographic algorithm [Digital Signature Algorithm] and cryptographic key sizes [1024 bits] that meet the following: [FIPS PUB 186-2].

FCS_COP.1 (2) Cryptographic operation

97 FCS_COP.1.1 - The TSF shall [perform encryption and decryption] in accordance with a specified cryptographic algorithm [NSA-certified high-robustness cryptography] and cryptographic key sizes [to be provided by NSA corresponding to the NSA-certified high-robustness algorithm] that meet the following: [NSA-provided standards].

FCS_COP.1 (3) Cryptographic operation

98 FCS_COP.1.1 - The TSF shall [compute a secure hash] in accordance with a specified cryptographic algorithm [Secure Hash Algorithm (SHA-1)] and cryptographic key sizes [not applicable] that meet the following: [FIPS PUB 180-1].

5.1.3 USER DATA PROTECTION (FDP)

FDP_ETC.1 Export of user data without security attributes

99 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.

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

FDP_ETC.2 Export of user data with security attributes

101 FDP_ETC.2.1 - The TSF shall enforce the [Mandatory Access Control SFP] when exporting user data, controlled under the SFP(s), outside of the TSC.

102 FDP_ETC.2.2 - The TSF shall export the user data with the user data's associated security attributes.

103 FDP_ETC.2.3 - The TSF shall ensure that the security attributes, when exported outside the TSC, are unambiguously associated with the exported user data.

104 FDP_ETC.2.4 - The TSF shall enforce the following rules when user data is exported from the TSC: [

a) If the SMTP or X.400 message has a security label, use the security label as the asserted classification; and

b) If the SMTP or X.400 message does not have a security label, use the classification field in the message header as the asserted classification].

FDP_IFC.1 Subset information flow control

105 FDP_IFC.1.1 The TSF shall enforce the [Mandatory Access Control SFP] on [

a) Subjects: processes operating in network enclaves;

b) Information: RFC 822 and X.400 mail messages; and

c) Operations: Information flow from one network enclave to another network enclave].

106 Application Note: With respect to the Mandatory Access Control SFP, a flow is equivalent to a write to the destination network enclave.

FDP_IFF.2 Hierarchical security attributes

107 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) Subject Security Attributes: Security level of the source network enclave; and

b) Information Security Attributes:

Security level of the destination network enclave;

Security level of the mail message (RFC 822 or X.400);

Sender status (restricted or unrestricted);

Recipient status (restricted or unrestricted);

Host status (restricted or unrestricted);

Destination status (restricted or unrestricted);

Attachment type;

Encrypted;

Digital Signature;

Dirty Word check; and

Other security attributes {to be determined by the Security Target writer}].

108 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 subject can read an object if the hierarchical classification in the subject's security level is greater than or equal to the hierarchical classification in the object's security level and the non-hierarchical categories in the subject's security level include all the non-hierarchical categories in the objects' security level. A subject can write an object only if the hierarchical classification in the subject's security level is less than or equal to the hierarchical classification in the object's security level and the non-hierarchical categories in the subject's security level are included in the non-hierarchical categories in the object's security level].

109 FDP_IFF.2.3 - The TSF shall enforce the [Additional Information Flow Control rule as follows: The Guard shall be configured to allow messages to flow from one network enclave at one security level to another network enclave at a potentially different security level (i.e., one for each source to destination network pair). The Guard shall be configured to ensure that electronic mail messages shall only flow between network enclaves under conditions that support the enforcement of the Mandatory Access Control SFP].

110 FDP_IFF.2.4 - The TSF shall provide the following [configurable security filters to support the Additional Information Flow Control rule].

111 FDP_IFF.2.5 - The TSF shall explicitly authorise an information flow based on the following rules:

a) [The TOE shall allow only received email messages that are composed entirely of allowable ASCII characters (i.e., non-ASCII email is prohibited) to pass through the TOE.

b) The TOE shall allow only email messages received from a classified enclave that contains a valid security label (i.e., improperly labeled email is prohibited) to pass through the TOE.

c) The TOE shall allow each received email message whose SMTP source identification data is not from a restricted email sender (i.e., restricted-source email is prohibited) to pass through the TOE.

d) The TOE shall allow each received email message whose SMTP destination identification data is not destined for one or more restricted email recipients (i.e., restricted destination email is prohibited) to pass through the TOE. A restricted email recipient is an email recipient on a network connected to the TOE who is not allowed to receive messages through the TOE. A restricted email recipient shall be a direct addressee (i.e., identified as a "TO:" recipient) or a courtesy copy addressee (i.e., identified as a "CC:" recipient).

e) The TOE shall allow each received email message whose SMTP source identification data is not from a restricted email host (i.e., restricted host email is prohibited) to pass through the TOE.

f) The TOE shall allow each received email message whose SMTP destination identification data is not from a restricted email host (i.e., restricted host email is prohibited) to pass through the TOE.

g) The TOE shall allow each received email message that includes only authorized attachments (i.e., unauthorized attachments are prohibited) to pass through the TOE. Specification of each attachment type shall include file characteristics associated with the attachment type.

h) The TOE shall allow only reviewed attachments (i.e., attachments that are not reviewed are prohibited) to pass through the TOE.

i) The TOE shall perform a dirty word search of received email messages and allow each message that passes the dirty word search (i.e., messages with dirty words are prohibited) to pass through the TOE.

j) The TOE shall allow encrypted messages to pass through the TOE.

k) The TOE shall allow signed messages to pass through the TOE.

l) The TOE shall allow messages that are both signed and encrypted to pass through the TOE.

m) Additional Mandatory Access Control SFP rules {to be determined by the Security Target writer}].

112 FDP_IFF.2.6 - The TSF shall explicitly deny an information flow based on the following rules: [

a) Protocols not supported by the TOE shall not be allowed to traverse the TOE.

b) The SMTP and X.400 portion of the TOE shall deny all communications other than electronic mail to pass through the TOE.

c) The X.500 and LDAP portion of the TOE shall deny all communications other than directory messages to pass through the TOE.]

113 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.

FDP_IFF.3 Limited illicit information flows

114 FDP_IFF.3.1 - The TSF shall enforce the [Mandatory Access Control SFP] to limit the capacity of [network-accessible illicit information flows] to a [ST assignment: maximum capacity].

115 Application Note: The ST author is expected to define the maximum capacity of all network-accessible illicit information flows and to provide an argument as to why each capacity is appropriate.

FDP_ITC.1 Import of user data without security attributes

116 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.

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

118 FDP_ITC.1.3 - The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [If a security label on RFC 822 or X.400 messages do not exist, the message shall be handled as though it was labeled with the security level of the network enclave from which it came].

FDP_ITC.2 Import of user data with security attributes

119 FDP_ITC.2.1 - The TSF shall enforce the [Mandatory Access Control SFP] when importing user data, controlled under the SFP, from outside of the TSC.

120 FDP_ITC.2.2 - The TSF shall use the security attributes associated with the imported user data.

121 FDP_ITC.2.3 - The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

122 FDP_ITC.2.4 - The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

123 FDP_ITC.2.5 - The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [If the security label of the message does not match the label in the RFC 822 or X.400 message, the TOE shall not release the message].

FDP_RIP.2 Full residual information protection

124 FDP_RIP.2.1 - The TSF shall ensure that any previous information content of a resource is made unavailable upon the *allocation of the resource to* all objects.

5.1.4 IDENTIFICATION AND AUTHENTICATION (FIA)

125 TOE security functions implemented by a probabilistic or permutation mechanism (e.g., password function) are required (at EAL2 and higher) to include a strength of function claim. The single-use authentication mechanism must demonstrate SOF-high. SOF-high is defined in Part 1 of the CC to be "a level of the TOE strength of function where analysis shows that the function provides adequate protection against deliberately, planned, or organized breach of TOE security by attackers possessing a high attack potential".

FIA_AFL.1 Authentication failure handling

126 FIA_AFL.1.1 - The TSF shall detect when [a settable, non-zero number {to be determined by the Security Target writer(s)}] **of** unsuccessful authentication attempts occur related to [user authentication].

127 FIA_AFL.1.2 - When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [prevent the identified user from successfully authenticating itself to the TOE until an action is taken by the Authorized Administrator].

FIA_ATD.1 User attribute definition

128 FIA_ATD.1.1 - The TSF shall maintain the following list of security attributes belonging to individual users: [identity, role associations, security clearance, and any other user security attributes {to be determined by the Security Target writer(s)}].

FIA_UAU.2 User authentication before any action

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

FIA_UAU.4 Single-use authentication mechanisms

130 FIA_UAU.4.1 – The TSF shall prevent reuse of authentication data related to [one-time passwords, digital certificates or biometrics].

FIA_UID.2 User identification before any action

131 FIA_UID.2.1 - The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.1.5 SECURITY MANAGEMENT (FMT)

FMT_MOF.1 (1) Management of security functions behaviour

132 FMT_MOF.1.1 - The TSF shall restrict the ability to *disable and enable* the functions [

a) Non-ASCII mail filter;
b) Improperly labeled mail filter;
c) Restricted sender mail filter;
d) Restricted recipient mail filter;
e) Restricted source host mail filter;
f) Restricted destination host mail filter;
g) Attachment mail filter;
h) Encryption mail filter;
i) Digital signature mail filter;
j) Encrypted and digitally signed mail filter;
k) Dirty word search mail filter; and
l) Other functions {to be determined by the Security Target writer}]

to [the Guard Application Administrator].

FMT_MOF.1 (2) Management of security functions behavior

133 FMT_MOF.1.1 - The TSF shall restrict the ability to *enable, disable, determine and modify the behavior of* the functions:

a) Security monitoring rules;
b) Actions to be taken in case of imminent audit storage failure;
c) Actions to be taken in the event of authentication failure;
d) Group of users assigned to a security role and their assigned functions;
e) Conditions under which abstract machine testing and self-test occurs;
f) Types of service failures handled;
g) List and actions for which replay is detected; and
h) Actions requiring trusted path;

to [an Authorized Administrator].

FMT_MSA.1 Management of security attributes

134 FMT_MSA.1.1 - The TSF shall enforce the [Mandatory Access Control SFP] to restrict the ability to [perform operations as specified in Table 3] the security attributes [as specified in Table 3] to [Guard Application Administrator].

Operation	Security Attribute
Change security label.	Security label of an RFC 822 or X.400 message
Change from restricted to unrestricted and vice versa. This shall be done by either specifying identification data for each restricted email sender or by specifying identification data for each unrestricted email sender. Any sender not identified as unrestricted is a restricted email sender.	Sender Status
Change from restricted to unrestricted and vice versa. This shall be done by either specifying identification data for each restricted email recipient or by specifying identification data for each unrestricted email recipient. Any recipient not identified as unrestricted is a restricted email recipient.	Recipient Status
Change from restricted to unrestricted and vice versa. This shall be done by either specifying identification data for each restricted source host or by specifying identification data for each unrestricted source host. Any sender not identified as unrestricted is a restricted source host.	Host Status
Change from restricted to unrestricted and vice versa. This shall be done by either specifying identification data for each restricted destination host or by specifying identification data for each unrestricted destination host. Any recipient not identified as unrestricted is a restricted destination host.	Destination Status

Operation	Security Attribute
Change allowed or prohibited attachment types.	Attachment Types
Change from encrypted, signed or both signed and encrypted.	Encrypted, Digital Signature, Encrypted and Digitally Signed
Turn on or off manual review.	Reviewed
Change dirty word list.	Dirty Word List

Table 3 - Management of Security Attributes

FMT_MSA.2 Secure security attributes

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

FMT_MSA.3 Static attribute initialisation

136 FMT_MSA.3.1 - The TSF shall enforce the [Mandatory Access Control SFP] to provide *restrictive* default values for security attributes that are used to enforce the SFP.

137 FMT_MSA.3.2 - The TSF shall allow the *Authorized Administrator* to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1 Management of TSF data

138 FMT_MTD.1.1 - The TSF shall restrict the ability to [perform operations as specified in Table 4] the [TOE data as specified in Table 4] to [the Authorized Administrator].

Operation	TOE Data
Specify	Identification data for a set of related email hosts (where every host in the set has the same status, either restricted or unrestricted) by specifying a single set of identification data; e.g., allow host identification data to include a "*" wildcard character (where "*" represents any combination of one or more characters).
Specify	File characteristics associated with an attachment type.
Query, Modify, Delete, and Assign	User attributes defined in FIA_ATD.1.
Set	Time and date used to form the timestamps in FPT_STM.1.
Query, Modify, Delete and	User identities in FIA_UID.2.

Operation	TOE Data
Assign	
Query, Modify, Delete and Assign	Authentication data in FIA_UAU.2.

Table 4 - Management of TOE Data

FMT_SMR.2 Restrictions on security roles

139 FMT_SMR.2.1 - The TSF shall maintain the roles: [Authorized Administrator, Guard Application Administrator, and other roles {to be determined by the Security Target writer(s)}].

140 FMT_SMR.2.2 - The TSF shall be able to associate users with roles.

141 FMT_SMR.2.3 - The TSF shall ensure that the conditions [Authorized Administrator and Guard Application Administrator functions are appropriately separated and a user authorized to exercise functions in one role can be prevented from exercising functions simultaneously in another role] are satisfied.

FMT_SMR.3 Assuming roles

142 FMT_SMR.3.1 - The TSF shall require an explicit request to assume the following roles: [Authorized Administrator, Guard Application Administrator, and other roles {to be determined by the Security Target writer(s)}].

5.1.6 PROTECTION OF THE TOE SECURITY FUNCTIONS (FPT)

FPT_AMT.1 Abstract machine testing

143 FPT_AMT.1.1 - The TSF shall run a suite of tests *during initial start-up, at the request of an authorised **Administrator**, and during automated recovery* to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

FPT_ITT.1 Basic internal TSF data transfer protection

144 FPT_ITT.1 – The TSF shall protect TSF data from *disclosure* when it is transmitted between separate parts of the TOE.

FPT_RCV.2 Automated recovery

145 FPT_RCV.2.1 - When automated recovery from a failure or service discontinuity is not possible, the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

146 FPT_RCV.2.2 - For [system failure and other failures {to be determined by the Security Target writer(s)}], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

FPT_RPL.1 Replay detection

147 FPT_RPL.1.1 - The TSF shall detect replay for the following entities: [Authorized Administrator and Guard Application Administrator authentication].

148 FPT_RPL.1.2 - The TSF shall perform [ignore the attempted replay operation and generate an audit record] when replay is detected.

FPT_RVM.1 Non-bypassability of the TSP

149 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.

FPT_SEP.2 SFP domain separation

150 FPT_SEP.2.1 - The unisolated portion of the TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

151 FPT_SEP.2.2 - The TSF shall enforce separation between the security domains of subjects in the TSC.

152 FPT_SEP.2.3 - The TSF shall maintain the part of the TSF related to [Mandatory Access Control SFP] in a security domain for their own execution that protects them from interference and tampering by the remainder of the TSF and by subjects untrusted with respect to those SFPs.

FPT_STM.1 Reliable time stamps

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

FPT_TDC.1 Inter-TSF basic TSF data consistency

154 FPT_TDC.1.1 – The TSF shall provide the capability to consistently interpret [security labels] when shared between the TSF and another trusted IT product.

155 FPT_TDC.1.2 – The TSF shall use [the following rule to interpret security labels: if the security label of the message does not match the label in the RFC 822 or X.400 classification field, the TOE shall not release the message] when interpreting the TSF data from another trusted IT product.

FPT_TST.1 TSF testing

156 FPT_TST.1.1 - The TSF shall run a suite of self tests *during initial start-up, at the request of the authorised **Administrator**,* **and** [during automated recovery] to demonstrate the correct operation of the TSF.

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

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

5.1.7 TRUSTED PATH (FTP)

FTP_ITC.1 Inter-TSF trusted channel

159 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.

160 FTP_ITC.1.2 – The TSF shall permit *the TSF* to initiate communication via the trusted channel.

161 FTP_ITC.1.3 – The TSF shall initiate communication via the trusted channel for [communication between the TOE and the cryptographic module].

FTP_TRP.1 Trusted path

162 FTP_TRP.1.1 - The TSF shall provide a communication path between itself and *local* **Authorized Administrator(s) and Guard Application Administrator(s)** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.

163 FTP_TRP.1.2 - The TSF shall permit *local **Authorized Administrator(s) and Guard Application Administrator(s)*** to initiate communication via the trusted path.

164 FTP_TRP.1.3 - The TSF shall require the use of the trusted path for *initial **Authorized Administrator(s) and Guard Application Administrator(s)** authentication* **and** [other services {to be determined by the Security Target writer}].

5.2 SECURITY REQUIREMENTS FOR THE ENVIRONMENT

FCS_CKM.1 Cryptographic key generation

165 FCS_CKM.1.1 - The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [NSA-certified high-robustness algorithm] and specified cryptographic key sizes [to be provided by NSA corresponding to the NSA-certified high-robustness algorithm] that meet the following: [NSA-provided standards].

FCS_CKM.2 Cryptographic key distribution

166 FCS_CKM.2.1 - The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [NSA-certified high-robustness method] that meets the following: [NSA-provided standards].

FCS_CKM.4 Cryptographic key destruction

167 FCS_CKM.4.1 - The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [NSA-certified high-robustness method] that meets the following: [NSA-provided standards].

5.3 TOE SECURITY ASSURANCE REQUIREMENTS

168 The TOE assurance requirements are EAL4 augmented by ADV_FSP.3, ADV_HLD.4, ADV_IMP.3, ADV_INT.2, ADV_LLD.2, ADV_RCR.2, ALC_DVS.2, ATE_COV.3, ATE_DPT.2, ATE_FUN.2, AVA_CCA.2, AVA_MSU.3, and AVA_VLA.4 as shown in Table 5.

Assurance Class	Assurance Components	
Configuration management	ACM_AUT.1	Partial CM automation
	ACM_CAP.4	Generation support and acceptance procedures
	ACM_SCP.2	Problem tracking CM coverage
Delivery and operation	ADO_DEL.2	Detection of modification
	ADO_IGS.1	Installation, generation, and start-up procedures
Development	ADV_FSP.3	Semiformal functional specification
	ADV_HLD.4	Semiformal high-level explanation
	ADV_IMP.3	Structured implementation of the TSF
	ADV_INT.2	Reduction of complexity
	ADV_LLD.2	Semiformal low-level design
	ADV_RCR.2	Semiformal correspondence demonstration
	ADV_SPM.2	Semiformal TOE security policy model
Guidance Documents	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
Life Cycle support	ALC_DVS.2	Sufficiency of security measures
	ALC_FLR.3	Systematic flaw remediation
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.1	Well-defined development tools
Tests	ATE_COV.3	Rigorous analysis of coverage

Assurance Class	Assurance Components	
	ATE_DPT.2	Testing: low-level design
	ATE_FUN.2	Ordered functional testing
	ATE_IND.2	Independent testing - sample
Vulnerability assessment	AVA_CCA.2	Systematic covert channel analysis
	AVA_MSU.3	Analysis and testing for insecure states
	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.4	Highly resistant

Table 5 - Security Assurance Requirements

5.3.1 CONFIGURATION MANAGEMENT (ACM)

ACM_AUT.1 Partial CM automation

Developer action elements:

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

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

Content and presentation of evidence elements:

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

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

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

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

Evaluator action elements:

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

ACM_CAP.4 Generation support and acceptance procedures

Developer action elements:

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

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

178 ACM_CAP.4.3D - The developer shall provide CM documentation.

Content and presentation of evidence elements:

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

180 ACM_CAP.4.2C - The TOE shall be labeled with its reference.

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

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

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

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

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

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

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

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

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

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

Evaluator action elements:

191 ACM_CAP.4.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ACM_SCP.2 Problem tracking CM coverage

Developer action elements:

192 ACM_SCP.2.1D - The developer shall provide CM documentation.

Content and presentation of evidence elements:

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

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

Evaluator action elements:

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

5.3.2 DELIVERY AND OPERATION (ADO)

ADO_DEL.2 Detection of modification

Developer action elements:

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

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

Content and presentation of evidence elements:

198 ADO_DEL.2.1C - The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

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

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

Evaluator action elements:

201 ADO_DEL.2.1E - 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

Developer action elements:

202 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:

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

Evaluator action elements:

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

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

5.3.3 DEVELOPMENT (ADV)

ADV_FSP.3 Semiformal functional specification

Developer action elements:

206 ADV_FSP.3.1D - The developer shall provide a functional specification.

Content and presentation of evidence elements:

207 ADV_FSP.3.1C - The functional specification shall describe the TSF and its external interfaces using a semiformal style, supported by informal, explanatory text where appropriate.

208 ADV_FSP.3.2C - The functional specification shall be internally consistent.

209 ADV_FSP.3.3C - The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

210 ADV_FSP.3.4C - The functional specification shall completely represent the TSF.

211 ADV_FSP.3.5C - The functional specification shall include rationale that the TSF is completely represented.

Evaluator action elements:

212 ADV_FSP.3.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

213 ADV_FSP.3.2E -The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

ADV_HLD.4 Semiformal high-level explanation

Developer action elements:

214 ADV_HLD.4.1D - The developer shall provide the high-level design of the TSF.

Content and presentation of evidence elements:

215 ADV_HLD.4.1C - The presentation of the high-level design shall be semiformal.

216 ADV_HLD.4.2C - The high-level design shall be internally consistent.

217 ADV_HLD.4.3C - The high-level design shall describe the structure of the TSF in terms of subsystems.

218 ADV_HLD.4.4C - The high-level design shall describe the security functionality provided by each subsystem of the TSF.

219 ADV_HLD.4.5C - 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.

220 ADV_HLD.4.6C - The high-level design shall identify all interfaces to the subsystems of the TSF.

221 ADV_HLD.4.7C - The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

222 ADV_HLD.4.8C - The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing complete details of all effects, exceptions and error messages.

223 ADV_HLD.4.9C - The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

224 ADV_HLD.4.10C - The high-level design shall justify that the identified means of achieving separation, including any protection mechanisms, are sufficient to ensure a clear and effective separation of TSP-enforcing from non-TSP-enforcing functions.

225 ADV_HLD.4.11C - The high-level design shall justify that the TSF mechanisms are sufficient to implement the security functions identified in the high-level design.

Evaluator action elements:

226 ADV_HLD.4.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

227 ADV_HLD.4.2E - The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_IMP.3 Structured implementation of the TSF

Developer action elements:

228 ADV_IMP.3.1D - The developer shall provide the implementation representation for the entire TSF.

Content and presentation of evidence elements:

229 ADV_IMP.3.1C - The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

230 ADV_IMP.3.2C - The implementation representation shall be internally consistent.

231 ADV_IMP.3.3C - The implementation representation shall describe the relationships between all portions of the implementation.

232 ADV_IMP.3.4C - The implementation representation shall be structured into small and comprehensible sections.

Evaluator action elements:

233 ADV_IMP.3.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

234 ADV_IMP.3.2E - The evaluator shall determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements.

ADV_INT.2 Reduction of complexity

Developer action elements:

235 ADV_INT.2.1D - The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

236 ADV_INT.2.2D - The developer shall provide an architectural description.

237 ADV_INT.2.3D - The developer shall design and structure the TSF in a layered fashion that minimises mutual interactions between the layers of the design.

238 ADV_INT.2.4D - The developer shall design and structure the TSF in such a way that minimises the complexity of the portions of the TSF that enforce any access control and/or information flow control policies.

Content and presentation of evidence elements:

239 ADV_INT.2.1C - The architectural description shall identify the modules of the TSF and shall specify which portions of the TSF enforce the access control and/or information flow control policies.

240 ADV_INT.2.2C - The architectural description shall describe the purpose, interface, parameters, and effects of each module of the TSF.

241 ADV_INT.2.3C - The architectural description shall describe how the TSF design provides for largely independent modules that avoid unnecessary interactions.

242 ADV_INT.2.4C - The architectural description shall describe the layering architecture.

243 ADV_INT.2.5C - The architectural description shall show that mutual interactions have been minimised, and justify those that remain.

244 ADV_INT.2.6C - The architectural description shall describe how the portions of the TSF that enforce any access control and/or information flow control policies have been structured to minimise complexity.

Evaluator action elements:

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

246 ADV_INT.2.2E - The evaluator shall determine that both the low-level design and the implementation representation are in compliance with the architectural description.

ADV_LLD.2 Semiformal low-level design

Developer action elements:

247 ADV_LLD.2.1D - The developer shall provide the low-level design of the TSF.

Content and presentation of evidence elements:

248 ADV_LLD.2.1C - The presentation of the low-level design shall be semiformal.

249 ADV_LLD.2.2C - The low-level design shall be internally consistent.

250 ADV_LLD.2.3C - The low-level design shall describe the TSF in terms of modules.

251 ADV_LLD.2.4C - The low-level design shall describe the purpose of each module.

252 ADV_LLD.2.5C - The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

253 ADV_LLD.2.6C - The low-level design shall describe how each TSP-enforcing function is provided.

254 ADV_LLD.2.7C - The low-level design shall identify all interfaces to the modules of the TSF.

255 ADV_LLD.2.8C - The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

256 ADV_LLD.2.9C - The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing complete details of all effects, exceptions and error messages.

257 ADV_LLD.2.10C - The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

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

259 ADV_LLD.2.2E - The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_RCR.2 Semiformal correspondence demonstration

Developer action elements:

260 ADV_RCR.2.1D - The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

Content and presentation of evidence elements:

261 ADV_RCR.2.1C - 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.

262 ADV_RCR.2.2C - For each adjacent pair of provided TSF representations, where portions of both representations are at least semiformally specified, the demonstration of correspondence between those portions of the representations shall be semiformal.

Evaluator action elements:

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

ADV_SPM.2 Semiformal TOE security policy model

Developer action elements:

264 ADV_SPM.2.1D - The developer shall provide a TSP model.

265 ADV_SPM.2.2D - The developer shall demonstrate correspondence between the functional specification and the TSP model.

Content and presentation of evidence elements:

266 ADV_SPM.2.1C - The TSP model shall be semiformal.

267 ADV_SPM.2.2C - The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

268 ADV_SPM.2.3C - The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

269 ADV_SPM.2.4C - The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

270 ADV_SPM.2.5C - Where the functional specification is at least semiformal, the demonstration of correspondence between the TSP model and the functional specification shall be semiformal.

Evaluator action elements:

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

5.3.4 GUIDANCE DOCUMENTS (AGD)

AGD_ADM.1 Administrator guidance

Developer action elements:

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

Content and presentation of evidence elements:

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

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

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

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

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

278 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.

279 AGD_ADM.1.7C - The administrator guidance shall be consistent with all other documentation supplied for evaluation.

280 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:

281 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

Developer action elements:

282 AGD_USR.1.1D - The developer shall provide user guidance.

Content and presentation of evidence elements:

283 AGD_USR.1.1C - The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

284 AGD_USR.1.2C - The user guidance shall describe the use of user-accessible security functions provided by the TOE.

285 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.

286 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.

287 AGD_USR.1.5C - The user guidance shall be consistent with all other documentation supplied for evaluation.

288 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:

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

5.3.5 LIFE CYCLE SUPPORT (ALC)

ALC_DVS.2 Sufficiency of security measures

Developer action elements:

290 ALC_DVS.2.1D - The developer shall produce development security documentation.

Content and presentation of evidence elements:

291 ALC_DVS.2.1C - The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

292 ALC_DVS.2.2C - The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

293 ALC_DVS.2.3C - The evidence shall justify that the security measures provide the necessary level of protection to maintain the confidentiality and integrity of the TOE.

Evaluator action elements:

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

295 ALC_DVS.2.2E - The evaluator shall confirm that the security measures are being applied.

ALC_FLR.3 Systematic flaw remediation

Developer action elements:

296 ALC_FLR.3.1D - The developer shall document the flaw remediation procedures.

297 ALC_FLR.3.2D - The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

298 ALC_FLR.3.3D - The developer shall designate one or more specific points of contact for user reports and inquiries about security issues involving the TOE.

Content and presentation of evidence elements:

299 ALC_FLR.3.1C - The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

300 ALC_FLR.3.2C - The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

301 ALC_FLR.3.3C - The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

302 ALC_FLR.3.4C - The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

303 ALC_FLR.3.5C - The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

304 ALC_FLR.3.6C - The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

305 ALC_FLR.3.7C - The flaw remediation procedures shall include a procedure requiring timely responses for the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw.

Evaluator action elements:

306 ALC_FLR.3.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_LCD.1 Developer defined life-cycle model

Developer action elements:

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

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

Content and presentation of evidence elements:

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

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

Evaluator action elements:

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

ALC_TAT.1 Well-defined development tools

Developer action elements:

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

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

Content and presentation of evidence elements:

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

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

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

Evaluator action elements:

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

5.3.6 TESTING (ATE)

ATE_COV.3 Rigorous analysis of coverage

Developer action elements:

318 ATE_COV.3.1D - The developer shall provide an analysis of the test coverage.

Content and presentation of evidence elements:

319 ATE_COV.3.1C - The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

320 ATE_COV.3.2C - The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

321 ATE_COV.3.3C - The analysis of the test coverage shall rigorously demonstrate that all external interfaces of the TSF identified in the functional specification have been completely tested.

Evaluator action elements:

322 ATE_COV.3.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_DPT.2 Testing: low-level design

Developer action elements:

323 ATE_DPT.2.1D - The developer shall provide the analysis of the depth of testing.

Content and presentation of evidence elements:

324 ATE_DPT.2.1C - The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and low-level design.

Evaluator action elements:

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

ATE_FUN.2 Ordered functional testing

Developer action elements:

326 ATE_FUN.2.1D - The developer shall test the TSF and document the results.

327 ATE_FUN.2.2D - The developer shall provide test documentation.

Content and presentation of evidence elements:

328 ATE_FUN.2.1C - The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

329 ATE_FUN.2.2C - The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

330 ATE_FUN.2.3C - 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.

331 ATE_FUN.2.4C - The expected test results shall show the anticipated outputs from a successful execution of the tests.

332 ATE_FUN.2.5C - The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

333 ATE_FUN.2.6C - The test documentation shall include an analysis of the test procedure ordering dependencies.

Evaluator action elements:

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

ATE_IND.2 Independent testing - sample

Developer action elements:

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

Content and presentation of evidence elements:

336 ATE_IND.2.1C - The TOE shall be suitable for testing.

337 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:

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

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

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

5.3.7 VULNERABILITY ASSESSMENT (AVA)

AVA_CCA.2 Systematic covert channel analysis

Developer action elements:

341 AVA_CCA.2.1D - The developer shall conduct a search for covert channels for each information flow control policy.

342 AVA_CCA.2.2D - The developer shall provide covert channel analysis documentation.

Content and presentation of evidence elements:

343 AVA_CCA.2.1C - The analysis documentation shall identify covert channels and estimate their capacity.

344 AVA_CCA.2.2C - The analysis documentation shall describe the procedures used for determining the existence of covert channels, and the information needed to carry out the covert channel analysis.

345 AVA_CCA.2.3C - The analysis documentation shall describe all assumptions made during the covert channel analysis.

346 AVA_CCA.2.4C - The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.

347 AVA_CCA.2.5C - The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.

348 AVA_CCA.2.6C - The analysis documentation shall provide evidence that the method used to identify covert channels is systematic.

Evaluator action elements:

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

350 AVA_CCA.2.2E - The evaluator shall confirm that the results of the covert channel analysis show that the TOE meets its functional requirements.

351 AVA_CCA.2.3E - The evaluator shall selectively validate the covert channel analysis through testing.

AVA_MSU.3 Analysis and testing for insecure states

Developer action elements:

352 AVA_MSU.3.1D - The developer shall provide guidance documentation.

353 AVA_MSU.3.2D - The developer shall document an analysis of the guidance documentation.

Content and presentation of evidence elements:

354 AVA_MSU.3.1C - The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

355 AVA_MSU.3.2C - The guidance documentation shall be complete, clear, consistent and reasonable.

356 AVA_MSU.3.3C - The guidance documentation shall list all assumptions about the intended environment.

357 AVA_MSU.3.4C - The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

358 AVA_MSU.3.5C - The analysis documentation shall demonstrate that the guidance documentation is complete.

Evaluator action elements:

359 AVA_MSU.3.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

360 AVA_MSU.3.2E - The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

361 AVA_MSU.3.3E - The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

362 AVA_MSU.3.4E - The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

363 AVA_MSU.3.5E - The evaluator shall perform independent testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

AVA_SOF.1 Strength of TOE security function evaluation

Developer action elements:

364 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:

365 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.

366 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:

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

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

AVA_VLA.4 Highly resistant

Developer action elements:

369 AVA_VLA.4.1D - The developer shall perform and document an analysis of the TOE deliverables searching for ways in which a user can violate the TSP.

370 AVA_VLA.4.2D - The developer shall document the disposition of identified vulnerabilities.

Content and presentation of evidence elements:

371 AVA_VLA.4.1C - The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

372 AVA_VLA.4.2C - The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

373 AVA_VLA.4.3C - The evidence shall show that the search for vulnerabilities is systematic.

374 AVA_VLA.4.4C - The analysis documentation shall provide a justification that the analysis completely addresses the TOE deliverables.

Evaluator action elements:

375 AVA_VLA.4.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

376 AVA_VLA.4.2E - The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

377 AVA_VLA.4.3E - The evaluator shall perform an independent vulnerability analysis.

378 AVA_VLA.4.4E - The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

379 AVA_VLA.4.5E - The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a high attack potential.

6.0 RATIONALE

380 This section describes the rationale for the Security Objectives and Security Functional Requirements as defined in Section 4 and Section 5, respectively. Additionally, this section describes the rationale for the Assurance Requirements; rationale for not satisfying all of the dependencies; and the rationale for the Strength of Function (SOF). Table 6 illustrates the mapping from Security Objectives to Threats and Policies. Table 7 illustrates the mapping of the Functional Requirements to Security Objectives.

6.1 RATIONALE FOR TOE SECURITY OBJECTIVES

O.ACCOUNTABILITY

381 This security objective is necessary to counter the threats: T.ADDRESS_SPOOFING and T.IDENTIFICATION_AUTHENTICATION because it requires that users be properly identified and held accountable for their actions (including sending messages) or when they use security functions. This security objective also necessitates that security-related events be associated with the identity of the user for purposes of auditing.

O.ADMIN_SUPPORT

382 This security objective is necessary to counter the threats: T.ADMINISTRATION and T.EXCESS_AUDIT which have to do with ensuring that Authorized Administrators and Guard Application Administrators have the proper administrative tools to effectively perform their duties, maintain the secure operation of the TOE and decrease the likelihood of administrative errors. This security objective necessitate that an action, required by the TOE, be taken prior to granting the 'role' of Authorized Administrator and Guard Application Administrator.

O.AUDIT

383 This security objective is necessary to counter the threats: T.AUDIT_FULL and T.AUDIT_UNDETECTED because it ensures that security-relevant events are detected and completely and accurately recorded. This security objective also ensures that the TOE detects when the audit log is approaching its capacity, and notifies the Authorized Administrator and/or the Guard Application Administrator accordingly.

O.AUDIT_PROTECT

384 This security objective is necessary to counter the threats: T.AUDIT_FULL and T.EXCESS_AUDIT because it ensures that the audit log is protected from deletion and modification.

O.AUDIT_SELECT

385 This security objective is necessary to counter the threat: T.EXCESS_AUDIT because it ensures that the Authorized Administrator is able to change the selection of auditable events during normal TOE operation.

O.AUTHENTICATION

386 This security objective is necessary to counter the threats: T.ADDRESS_SPOOFING, T.BRUTE_FORCE and T.IDENTIFICATION_AUTHENTICATION because it requires that users are uniquely identified via a single-use authentication mechanism and only granted a limited number of authentication attempts prior to accessing the TOE.

O.CONFIDENTIALITY

387 This security objective is necessary to counter the threats and policy: T.DISCLOSURE, T.INCORRECT_LEVEL and P.CRYPTOGRAPHY because it requires that the TOE utilizes encryption and employs cryptography of adequate strength to protect messages and data from unauthorized disclosure.

O.COVERT_CHANNEL

388 This security objective is necessary to counter the threat: T.COVERT_CHANNEL because it requires that the type and capacity of illicit information flows are limited.

O.CRYPTOGRAPHY

389 This security objective is necessary to counter the threat and policy T.CRYPTOGRAPHIC_ATTACK and P.CRYPTOGRAPHY because it ensures that the cryptography used in the TOE is compliant with the GIG.

O.DATA_INTEGRITY

390 This security objective is necessary to counter the threat: T.MODIFY_DATA because it requires that messages and security-related data are protected from unauthorized modification.

O.DOMAIN_SEPARATION

391 This security objective is necessary to counter the threat: T.MODIFY_DATA because it ensures that the TOE is resistant to interference, modification or destruction by unauthorized external sources and that its domain is strictly maintained for execution.

O.IMPERSONATE

392 This security objective is necessary to counter the threats: T.MASQUERADE and T.REPLAY because it requires that a trusted path be established between the user and the TOE when entering authentication information. Additionally, it ensures that all digital signatures are validated.

O.INFORMATION_FLOW

393 This security objective is necessary to counter the threats and policy: T.INCORRECT_LEVEL, T.SECURITY_LEVEL and P.MANDATORY_ACCESS_CONTROL because it ensures that information residing on the TOE is not released from a higher-level enclave to an enclave containing a lower security level or between non-comparable security levels. This security objective also ensures that the TOE is able to correctly associate a security level with data upon import or export.

O.MULTI-LEVEL_PORT

394 This security objective is necessary to counter the threat: T.SECURITY_LEVEL which has to do with correctly interpreting security labels on messages that are imported into or exported from the TOE.

O.NON-BYPASSABILITY

395 This security objective is necessary to counter the threats: T.BYPASS, T.DISCLOSURE and T.HIGH_ATTACK_POTENTIAL because it requires that the TOE is always invoked and that messages are not releasable until the security enforcement functions are invoked and successful.

O.RECOVERY

396 This security objective is necessary to counter the threat: T.SYSTEM_FAILURE because it requires that the TOE automatically recovers to a secure state upon the event of a system failure or discontinuity of operation.

O.ROLE_SEPARATION

397 This security objective is necessary to counter the threat: T.IDENTIFICATION_AUTHENTICATION because it requires that there be separate roles for the Authorized Administrator and the Guard Application Administrator and that there are rules that control the relationship between the roles.

O.SELF_PROTECT

398 This security objective is necessary to counter the threats: T.BYPASS, T.MODIFY_DATA, T.SYSTEM_FAILURE and T.HIGH_ATTACK_POTENTIAL because it requires that the TOE protect itself from attempts to bypass, modify, destroy or tamper with TOE security-critical TOE data or programs.

O.SELF_TEST

399 This security objective is necessary to counter the threats: T.MODIFY_DATA and T.SYSTEM_FAILURE because it requires the TOE to execute a suite of self tests during initial startup, upon request by the Authorized Administrator and during automated recovery (i.e., in the event of a system failure) to ensure the integrity of the TOE code and its data structures.

O.SINGLE_LEVEL_PORT

400 This security objective is necessary to counter the threat: T.SECURITY_LEVEL because it ensures that the TOE attaches a security label to all unlabeled messages/data entering the TOE (i.e., the label must be equal to the level of the source address) or exiting the TOE (i.e., the label must equal the level of the TOE).

O.SOF

401 This security objective is necessary to counter the threat: T.HIGH_ATTACK_POTENTIAL because it requires that the TOE is resistant to penetration attacks performed by a threat agent possessing a high attack potential.

	O.ACCOUNTABILITY	O.ADMIN_SUPPORT	O.AUDIT	O.AUDIT_PROTECT	O.AUDIT_SLELECT	O.AUTHENTICATION	O.CONFIDENTIALITY	O.COVERT_CHANNEL	O.CRYPTOGRAPHY	O.DATA_INTEGRITY	O.DOMAIN_SEPARATION	O.IMPERSONATE	O.INFORMATION_FLOW	O.MULTI_LEVEL_PORT	O.NON-BYPASSIBLITY	O.RECOVERY	O.ROLE_SEPARATION	O.SELF_PROTECT	O.SELF_TEST	O.SINGLE_LEVEL_PORT	O.SOF
T.ADDRESS_SPOOFING	X					X															
T.ADMINSTRATION		X																			
T.AUDIT_FULL			X	X																	
T.AUDIT_UNDETECTED			X																		
T.BRUTE_FORCE						X															
T.BYPASS															X			X			
T.COVERT_CHANNEL								X													
T.CRYPTOGRAPHIC_ATTACK									X												
T.DISCLOSURE							X								X						
T.EXCESS_AUDIT		X		X	X																
T.HIGH_ATTACK_POTENTIAL															X			X			X
T.IDENTIFICATION_AUTHENTICATION	X					X											X				
T.INCORRECT_LEVEL							X						X								
T.MASQUERADE												X									
T.MODIFY_DATA										X	X							X	X		
T.REPLAY												X									
T.SECURITY_LEVEL													X	X						X	
T.SYSTEM_FAILURE																X		X	X		
P.CRYPTOGRAPHY							X		X												
P.MANDATORY_ACCESS_CONTROL													X								

Table 6 - Security Objectives to Threats/Policies Mapping

6.2 RATIONALE FOR SECURITY OBJECTIVES/REQUIREMENTS FOR THE ENVIRONMENT

402 All of the security objectives for the environment except one (i.e., O.KEY_PROTECTION) are restatements of assumptions found in Section 3. Therefore, those security objectives for the environment trace to the assumptions trivially. The environmental objective O.KEY_PROTECTION is necessary to counter the environmental threat T.KEY_COMPROMISE because it ensures the confidentiality and integrity of keys. Additionally, the environmental requirements FCS_CKM.1, FCS_CKM.2 and FCS_CKM.4 are necessary to ensure that the keys and key management data generated are of adequate strength to protect the confidentiality and integrity of electronic mail transmitted to/from the TOE, are distributed securely to provide confidentiality and integrity of electronic mail transmitted to/from the TOE, and are correctly destroyed to protect the confidentiality and integrity of electronic mail transmitted to/from the TOE. The Guard itself is not responsible for the management of keys. Specifically, the Guard does not generate, distribute or destroy keys. However, it is important that the Guard be integrated with another IT product or TOE that provides this functionality.

6.3 RATIONALE FOR SECURITY REQUIREMENTS

403 The functional and assurance requirements presented in this PP are mutually supportive and their combination meets the stated security objectives. The security requirements were derived according to the general model presented in Part 1 of the Common Criteria. Table 6 demonstrates the relationship among the threats, policies and TOE security objectives. Table 7 demonstrates the mapping between the security requirements and the security objectives. Together these tables demonstrate the completeness and sufficiency of the security requirements.

FAU_GEN.1 Audit Data Generation

404 This component outlines the data that must be included in audit records and the events that must be audited. This component traces back to and aids in meeting the following objective: O.AUDIT.

FAU_SAA.1 Potential Violation Analysis

405 This component ensures that repeated failed attempts to authenticate are monitored and alarmed if a threshold is reached. This component traces back to and aids in meeting the following objective: O.AUTHENTICATION.

FAU_SEL.1 Selective Audit

406 This component ensures that the Authorized Administrator and/or Guard Application Administrator can dynamically change the set of events to be audited. This component traces back to and aids in meeting the following objectives: O.ADMIN_SUPPORT and O.AUDIT_SELECT.

FAU_STG.1 Protected Audit Trail Storage

407 This component ensures that the audit trail is always protected from tampering. This component traces back to and aids in meeting the following objective: O.AUDIT_PROTECT.

FAU_STG.3 Action in Case of Possible Audit Data Loss

408 This component ensures that the Authorized Administrator and/or the Guard Application Administrator are notified when the audit trail is reaching its maximum capacity. This component traces back to and aids in meeting the following objective: O.AUDIT.

FAU_STG.4 Prevention of Audit Data Loss

409 This component ensures that the Authorized Administrator will be able to administer the audit trail should it become full. This component traces back to and aids in meeting the following objective: O.AUDIT.

FCS_COP.1 Cryptographic Operation

410 This component ensures that electronic mail sent to/from the TOE is encrypted using an NSA-certified cryptographic algorithm and signed using FIPS-approved digital signature and secure message hash algorithms. This component traces back to and aids in meeting the following objectives: O.CONFIDENTIALITY, O.CRYPTOGRAPHY, O.DATA_INTEGRITY, and O.IMPERSONATE.

FDP_ETC.1 Export of User Data without Security Attributes

411 This component ensures that the TOE processes unlabeled electronic mail messages securely. This component traces back to and aids in meeting the following objective: O.SINGLE_LEVEL.PORT.

FDP_ETC.2 Export of User Data with Security Attributes

412 This component ensures that the TOE processes labeled electronic mail messages securely. This component traces back to and aids in meeting the following objective: O.MULTI_LEVEL_PORT.

FDP_IFC.1 Subset Information Flow Control

413 This component identifies the entities involved in the Mandatory Access Control SFP. This component traces back to and aids in meeting the following objective: O.INFORMATION_FLOW.

FDP_IFF.2 Hierarchical Security Attributes

414 This component identifies the attributes of the subjects sending and receiving the information in the Mandatory Access Control SFP, as well as the attributes for the information itself. Then the operations identify under what conditions information is permitted to flow through the TOE. This component traces back to and aids in meeting the following objective: O.INFORMATION_FLOW.

FDP_IFF.3 Limited Illicit Information Flows

415 This component ensures that certain types of illicit information flows are limited to an acceptable capacity. This component traces back to and aids in meeting the following objective: O.COVERT_CHANNEL.

FDP_ITC.1 Import of User Data without Security Attributes

416 This component ensures that the TOE imports unlabeled electronic mail messages securely. This component traces back to and aids in meeting the following objective: O.SINGLE_LEVEL.PORT.

FDP_ITC.2 Import of User Data with Security Attributes

417 This component ensures that the TOE imports labeled electronic mail messages securely. This component traces back to and aids in meeting the following objective: O.MULTI_LEVEL_PORT.

FDP_RIP.2 Subset Residual Information Protection

418 This component ensures that all electronic mail that has traversed through the TOE and all TOE internal data are inaccessible after deletion. This component traces back to and aids in meeting the following objective: O.CONFIDENTIALITY.

FIA_AFL.1 Authentication Failure Handling

419 This component ensures that human users who are not Authorized Administrators or Guard Application Administrators cannot endlessly attempt to authenticate. After some number of failures, defined by the Authorized Administrator, the user is unable from that point on to authenticate. This component traces back to and aids in meeting the following objective: O.AUTHENTICATION.

FIA_ATD.1 User Attribute Definition

420 This component exists to provide attributes to distinguish Authorized Administrators and Guard Application Administrators from one another for accountability purposes and to associate the roles in FMT_SMR.2 with a user. This component traces back to and aids in meeting the following objective: O.ACCOUNTABILITY.

FIA_UAU.2 User Authentication Before Any Action

421 This component ensures that the users are authenticated before any action is allowed by the TSF. This component traces back to and aids in meeting the following objective: O.AUTHENTICATION.

FIA_UAU.4 Single-use Authentication Mechanisms

422 This component was chosen to ensure that Authorized Administrators and Guard Application Administrators use an authentication mechanism of adequate strength when authenticating to the TOE. This component traces back to and aids in meeting the following objective: O.AUTHENTICATION.

FIA_UID.2 User Identification Before Any Action

423 This component ensures that the users are identified to the TOE before anything occurs on behalf of the user. This component traces back to and aids in meeting the following objective: O.ACCOUNTABILITY.

FMT_MOF.1 (1) Management of Security Functions Behavior

424 This component ensures that the TOE restricts the ability to enable, disable, and modify the security filters to the Guard Application Administrator. This component traces back to and aids in meeting the following objective: O.ADMIN_SUPPORT.

FMT_MOF.1 (2) Management of Security Functions Behavior

425 This component ensures that the TOE restricts the ability to modify the behavior of functions (e.g., security monitoring rules; actions to be taken in case of imminent audit storage failure; actions to be taken in the event of authentication failure; group of users assigned to a security role and their assigned functions; conditions under which abstract machine testing and self-test occurs; types of service failures handled; list and actions for which replay is detected; and actions requiring trusted path) to the Authorized Administrator and/or Guard Application Administrator. This component traces back to and aids in meeting the following objective: O.ADMIN_SUPPORT.

FMT_MSA.1 Management of Security Attributes

426 This component ensures that the TSF restricts the ability to add, delete, and modify the security attributes that affect the Mandatory Access Control SFP to only the Authorized Administrator and/or Guard Application Administrator. This component traces back to and aids in meeting the following objectivs: O.ADMIN_SUPPORT.

FMT_MSA.2 Secure Security Attributes

427 This component ensures that appropriate values are assigned to the security attributes used in the Mandatory Access Control SFP. This component traces back to and aids in meeting the following objective: O.ADMIN_SUPPORT.

FMT_MSA.3 Static Attribute Initialization

428 This component ensures that there are restrictive default values implemented in the Mandatory Access Control SFP that the Authorized Administrator and/or Guard Application Administrator can change. This component traces back to and aids in meeting the following objective: O.SELF_PROTECT.

FMT_MTD.1 Management of TSF Data

429 This component ensures that the TSF restricts the ability to modify, delete, and assign user attributes (as defined in FIA_ATD.1.1), user identities (as defined in FIA_UID.2), authentication data (as defined in FIA_UAU.2) and timestamps (as defined in FPT_STM.1) to only the Authorized Administrator and/or the Guard Application Administrator. This component traces back to and aids in meeting the following objective: O.ADMIN_SUPPORT.

FMT_SMR.2 Restrictions on Security Roles

430 This component was chosen because each of the FMT components depends on the assignment of a user to the Authorized Administrator and Guard Application Administrator roles. This component traces back to and aids in meeting the following objective: O.ROLE_SEPARATION.

FMT_SMR.3 Assuming Roles

431 This component ensures that users must take an explicit action in order to assume a trusted role (i.e., Authorized Administrator or Guard Application Administrator). This component traces back to and aids in meeting the following objective: O.ADMIN_SUPPORT.

FPT_AMT.1 Underlying Abstract Machine Test

432 This component ensures that the security assumptions provided by the underlying abstract machine are tested during start-up. This component traces back to and aids in meeting the following objective: O.SELF_PROTECT.

FPT_RCV.2 Automated Recovery

433 This component ensures that the TOE returns to a secure state in the event of system failure. This component traces back to and aids in meeting the following objective: O.RECOVERY.

FPT_RPL.1 Replay Detection

434 This component ensures that replay of authentication attempts are detected and disallowed. This component traces back to and aids in meeting the following objectives: O.AUTHENTICATION and O.IMPERSONATE.

FPT_RVM.1 Non-bypassability of the TSP

435 This component ensures that the TOE enforcement functions are always invoked from initial start-up. This component traces back to and aids in meeting the following objective: O.NON_BYPASSABILITY.

FPT_SEP.2 SFP Domain Separation

436 This component ensures that the TSF has a domain of execution that is separate and that cannot be violated by unauthorized users. This component traces back to and aids in meeting the following objective: O.DOMAIN_SEPARATION.

FPT_STM.1 Reliable Time Stamps

437 This component was included because FAU_GEN.1 depends on having the date and time accurately recorded in the audit records. This component traces back to and aids in meeting the following objective: O.AUDIT.

FPT_TDC.1 Inter-TSF TSF Data Consistency

438 This component ensures that security labels sent between the TOE and another trusted IT product are interpreted correctly. This component traces back to and aids in meeting the following objective: O.MULTI_LEVEL_PORT.

FPT_TST.1 TSF Testing

439 This component ensures the integrity of the operation of the TSF and to provide the Authorized Administrator a means to verify the integrity of the TSF code and data. This component traces back to and aids in meeting the following objective: O.SELF_TEST.

FPT_ITT.1 Basic Internal TSF Data Transfer Protection

440 This component ensures that the cryptographic keys and data transmitted between different parts of the TOE are not disclosed. This component traces back to and aids in meeting the following objective: O.CONFIDENTIALITY.

FTP_ITC.1 Inter-TSF Trusted Channel

441 This component ensures that a trusted channel exists between users of the TOE and the cryptographic module. This component traces back to and aids in meeting the following objective: O.IMPERSONATE.

FTP_TRP.1 Trusted Path

442 This component ensures that a trusted path is available to users, giving them assurance that they are communicating with the TOE. This component traces back to and aids in meeting the following objective: O.IMPERSONATE.

443 A summary of the security requirements to security objectives mapping is contained in Table 7 below.

	O.ACCOUNTABILITY	O.ADMIN_SUPPORT	O.AUDIT	O.AUDIT_PROTECT	O.AUDIT_SELECT	O.AUTHENTICATION	O.CONFIDENTIALITY	O.COVERT_CHANNEL	O.CRYPTOGRAPHY	O.DATA_INTEGRITY	O.DOMAIN_SEPARATION	O.IMPERSONATE	O.INFORMATION_FLOW	O.MULTI_LEVEL_PORT	O.NON-BYPASSIBLITY	O.RECOVERY	O.ROLE_SEPARATION	O.SELF_PROTECT	O.SELF_TEST	O.SINGLE_LEVEL_PORT
FAU_GEN.1			X																	
FAU_SAA.1						X														
FAU_SEL.1		X			X															
FAU_STG.1				X																
FAU_STG.3			X																	
FAU_STG.4			X																	
FCS_COP.1							X		X	X		X								
FDP_ETC.1																				X
FDP_ETC.2														X						
FDP_IFC.1													X							
FDP_IFF.2													X							
FDP_IFF.3								X												
FDP_ITC.1																				X
FDP_ITC.2														X						
FDP_RIP.2							X													
FIA_AFL.1						X														
FIA_ATD.1	X																			
FIA_UAU.2						X														
FIA_UAU.4						X														

	O.ACCOUNTABILITY	O.ADMIN_SUPPORT	O.AUDIT	O.AUDIT_PROTECT	O.AUDIT_SELECT	O.AUTHENTICATION	O.CONFIDENTIALITY	O.COVERT_CHANNEL	O.CRYPTOGRAPHY	O.DATA_INTEGRITY	O.DOMAIN_SEPARATION	O.IMPERSONATE	O.INFORMATION_FLOW	O.MULTI_LEVEL_PORT	O.NON-BYPASSIBLITY	O.RECOVERY	O.ROLE_SEPARATION	O.SELF_PROTECT	O.SELF_TEST	O.SINGLE_LEVEL_PORT
FIA_UID.2	X																			
FMT_MOF.1 (1)		X																		
FMT_MOF.1 (2)		X																		
FMT_MSA.1		X																		
FMT_MSA.2		X																		
FMT_MSA.3																		X		
FMT_MTD.1		X																		
FMT_SMR.2																	X			
FMT_SMR.3		X																		
FPT_AMT.1																		X		
FPT_ITT.1							X													
FPT_RCV.2																X				
FPT_RPL.1						X						X								
FPT_RVM.1															X					
FPT_SEP.2											X									
FPT_STM.1			X																	
FPT_TDC.1														X						
FPT_TST.1																			X	
FTP_ITC.1												X								
FTP_TRP.1												X								

Table 7 - Functional Requirements to Security Objectives Mapping

6.4 RATIONALE FOR ASSURANCE REQUIREMENTS

444 EAL4 Augmented was chosen to ensure a high-level of confidence in the security services used to protect information in DoD Mail Guards for high-robustness environments. The assurance selection was based on:

- Detailed conversations with the sponsor of the PP;
- Recommendations documented in the GIG;
- The required strength of function, SOF-high (Section 4.1)
- EAL requirements as specified in the *Preferred Assurance Components/Processes for Devices Protecting Classified Information* table (Reference Appendix C); and
- The postulated threat environment (Section 3.3).

445 The sponsor of this PP determined that certain security critical components of the Mail Guard may require an EAL of greater than 4[4] to ensure that the security engineering performed by the developer was based on rigorous development practices supported by specialized security engineering techniques, such as the use of a structured development process, development of environment controls, comprehensive configuration management and evidence of secure product delivery. The Government's guidance in the GIG policy was consulted and found to also support the chosen assurance level[5].

446 In order to ensure the security of a high-assurance system, not only must vulnerability analysis be performed by the developer, but the NSA evaluator, through the use of independent functional testing, must search for vulnerabilities and demonstrate that they are highly resistant to penetration attackers with high attack potential (T.HIGH_ATTACK_POTENTIAL). This level of testing is supported by requirements ATE_FUN.2, ATE_COV.3, and ATE_DPT.2.

447 Since the threat to high robustness Mail Guard systems require greater protection, more detailed product information is required as indicated by requirements ADV_FSP.3, ADV_HLD.3, ADV_IMP.3, ADV_INT.2, ADV_LLD.2, ADV_RCR.2, ADV_SPM.2 and AVA.MSU.3 in this PP. To provide the necessary support to fielded Mail Guard systems, flaw remediation, ALC_FLR.3, augments the requirements for EAL4. The developer shall provide a mechanism to track and correct security flaws in the TOE that are discovered after initial delivery and installation. Additionally, the developer must provide for automatic distribution of security flaw reports and corrections to registered users that may be affected by the defect.

---

[4] Reference the Preferred Assurance Components as specified in Appendix C. The sponsor of this PP provided the EAL table to specify the necessary assurances for products or devices used to protect classified information.

[5] High robustness security services and mechanisms provide the most stringent protection and rigorous security countermeasures. High robustness solutions require high assurance security design, such as specified by NSA or the International Common Criteria (CC), at a minimum, to be an EAL greater than 4.

448 Lastly, the NSA evaluator must validate the developer's systematic covert channel analysis specified by requirement AVA_CCA.2, to confirm the non-existence of illicit information flows that may be exploited by threat agents possessing high attack potential.

6.5 RATIONALE FOR NOT SATISFYING ALL DEPENDENCIES

449 The FDP_IFC.1 dependency (i.e., FDP_IFF.1) is not included in this PP. This dependency is satisfied in this PP with the inclusion of FDP_IFF.2, which is hierarchical to FDP_IFF.1. The FDP_IFF.3 dependency of the assurance requirement AVA_CCA.1 is not included in this PP. This PP contain assurance requirement AVA_CCA.2 which requires that covert channel analysis be performed in a systematic way (i.e., structured and repeatable), as oppose to an ad-hoc analysis. Therefore, a greater level of assurance is achieved through assurance requirement AVA_CCA.2.

450 The functional requirements FMT_MOF.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1 and FMT_SMR.2 are dependent on the requirement FMT_SMR.1 (Security Roles). Since there are multiple roles within this PP (i.e., Authorized Administrator and Guard Application Administrator), it is required that the conditions or rules that control the relationship between these roles are specified. Therefore, functional requirement FMT_SMR.2 is included and is hierarchical to FMT_SMR.1. As such, the requirement FMT_SMR.1 is satisfied.

451 The FPT_RCV.2 has the assurance dependency of ADV_SPM.1, Informal TOE Security Policy Model as a dependency. According to Part 3 of the CC, the informal security policy model (ADV_SPM.1) and the semiformal TOE security policy model (ADV_SPM.2) are considered to be hierarchical in nature. Therefore, with the inclusion of ADV_SPM.2 in this PP, ADV_SPM.1 is satisfied.

6.6 RATIONALE FOR STRENGTH OF FUNCTION CLAIM

452 Part 1 of the CC defines the "strength of function" in terms of the minimum efforts assumed necessary to defeat the expected security behavior of a TOE security function. There are three strength of function levels defined in Part 1: SOF-basic, SOF-Medium and SOF-high. SOF-high is the strength of function level chosen for this PP. SOF-high states, "a level of the TOE strength of function where analysis shows that the function provides adequate protection against deliberately planned or organized breach of TOE security by attackers possessing a high attack potential". The rationale for choosing SOF-high was based on the TOE security objectives documented in Section 4 of this PP. Additionally, the sponsor determined that the SOF-high level is vital to address the TOE security objectives that counter the threat T.HIGH_ATTACK_POTENTIAL. Consequently, the metrics (i.e., password and keys) chosen for inclusion in this PP were determined to be sufficient for SOF-high and would adequately protect data and messages in a High Robustness Environment.

APPENDIX A: ACRONYMS

CC	Common Criteria for IT Security Evaluation
CM	Configuration Management
DOD	Department of Defense
DSA	Directory Service Agent
DUA	Directory User Agent
EAL	Evaluation Assurance Level
GIG	DoD Global Information Grid Information Assurance
HRE	High Robustness Environment
IP	Internet Protocol
IT	Information Technology
MAC	Mandatory Access Control
MTA	Mail Transfer Agent
MTS	Mail Transfer System
NSA	National Security Agency
PP	Protection Profile
SFP	Security Function Policy
SMTP	Simple Mail Transfer Protocol
SOF	Strength of Function
TOE	Target of Evaluation
TSE	TOE Security Environment
TSF	TOE Security Functions
TSP	TOE Security Policy

APPENDIX B: REFERENCES

[1] *Common Criteria for Information Technology Security Evaluation*, CCIB-99-031 through 033, Parts 1 – 3, Version 2.1, August 1999.

[2] *Defense Information Infrastructure (DII) High Assurance Mail Guard Protection Profile*, Version 0.3, August 1998.

[3] Department of Defense Chief Information Officer Guidance and Policy Memorandum No. 6-8510, *Guidance and Policy for the Department of Defense Global Information Grid Information Assurance (GIG)*, June 2000.

[4] Draft Department of Defense Instruction, *IA Implementation 8500.bb*, June 7, 2001.

[5] *Information Assurance Technical Framework*, Version 3.0, September 2000.

[6] *DII Guard Statement of Work (SOW),* Rev. J.2, 7 May 2001.

[8] *Secure Hash Standard*, FIPS Pub 180-1, April 1995.

[9] *Digital Signature Standard (DSS*), FIPS PUB 186-2, January 2000.

[10] *U. S. DoD Remote Access Protection Profile for SBU High Environments*, Version 0.9, May 2000.

[12] *Simple Mail Transfer Protocol (SMTP*), RFC 821, August 1982.

[13] *Multipart Internet Mail Extensions (MIME): Mechanisms for Specifying and Describing the Format of Internet Message Bodies,* RFC 1341, June 1992.

APPENDIX C: EAL TABLE

~~Preferred Assurance Components/Processes for Devices Protecting Classified Information - EAL*~~

Summary of Assurance Components by Evaluation Assurance Level

Assurance Class	Assurance Family	Assurance Components by Evaluation Assurance Level							
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7	
Configuration Management	**ACM_AUT**				1	1	2	2	Automation
	ACM_CAP	1	2	3	4	4	5	5	Prevents Unauthorized Modification
	ACM_SCP			1	2	3	3	3	Tracking Changes & Security Flaws
Delivery & Operation	**ADO_DEL**		1	1	2	2	2	3	Detects Modification During Delivery
	ADO_IGS	1	1	1	1	1	1	1	Provide Guidance for Installation/Start-Up
Development	**ADV_FSP**	1	1	1	2	3	3	4	Semi-formal Security Policy
	ADV_HLD		1	2	2	3	4	5	Semi-formal High Level Descriptions
	ADV_IMP				1	2	3	3	Explain Functions & Dependencies
	ADV_INT					1	2	3	Internal Design Requires Modularity & Layering
	ADV_LLD				1	1	2	2	Semi-formal Low Level Design
	ADV_RCR	1	1	1	1	2	2	3	Documentation Matches
	ADV_SPM				1(2)*	3	3	3	Semi-formal Security Policy Modeling
Guidance Documents	**AGD_ADM**	1	1	1	1	1	1	1	Guidance for Administration
	AGD_USR	1	1	1	1	1	1	1	Guidance for Users
Life Cycle Support	**ALC_DVS**			1	1	1	2	2	Controlled Development Process
	ALC_FLR				(3)*				Systematic Flaw Handling
	ALC_LCD				1	2	2	3	Must Have Life Cycle Model
	ALC_TAT				1	2	3	3	Basic Requirements for Development Tools
Tests	**ATE_COV**		1	2	2	2	3	3	Testing Philosophy & Procedures
	ATE_DPT			1	1	2	2	3	High Level Design Testing
	ATE_FUN		1	1	1	1	2	2	Functional Testing
	ATE_IND	1	2	2	2	2	2	3	Tester Must Duplicate Some Vendor Tests
Vulnerability Assessment	**AVA_CCA**					1	2	2	Non-exhaustive Covert Channel Analysis
	AVA_MSU			1	2	2	3	3	Misuse Analysis
	AVA_SOF		1	1	1	1	1	1	Strength of Function Analysis
	AVA_VLA		1	1	2	3	4	4	Thorough Analysis for Vulnerabilities

20-Nov-00

30-Jan-01