User-Oriented Protection Profile for Unobservable Message Delivery using MIX networks, $Revision: 2.4 $

Giovanni Iachello

$Date: 1999/06/06 17:46:55 $

Contents

List of Tables

List of Figures

1 Introduction

1.0.0.1 Protection Profile title: User-Oriented Protection Profile for Unobservable Message Delivery using MIX networks, $Revision: 2.4 $

1.0.0.2 Criteria Version: CC 15408-3 FDIS[JTC98a], with additional components from [Iac99c].

1.0.0.3 Protection Profile Label: (none)

1.0.0.4 Author: Giovanni Iachello

1.0.0.5 Keywords: anonymity, MIX, email

1.1 Foreword

This Protection Profile describes an unobservable message delivery application implemented by using MIX networks (for a broad discussion see [Cha81] and [FGJP98]), and states the security requirements as seen by the user of such a system. The functional and assurance requirements specification are written following the conventions set by the Common Criteria for Information Technology Security Evaluation (see [JTC98a]) and are in the form of a Protection Profile.

This document was made necessary after close examination of the Single MIX Protection Profile (see [Iac99b]) and Protection Profile for an Unobservable Message Delivery Application (see [Iac99a]), because of the difficulties of specifying security requirements protecting two different subjects, which could be antagonistic in nature, in one single document.

The two mentioned PPs have the common goal of protecting the operation of the Target Of Evaluation (TOE for short) from external attackers, and are devised using a TOE-centric approach. On the contrary, this PP examines the security problems of the user's perspective, and imposes conditions on the TOE operation and security functions, to protect the user from the threats against his assets. As a consequence, the following discussion on assets, threats and attackers, will focus on the user as entity to protect, and will overlook all the problems related to the protection of the MIX network.

1.2 A note on the terminology

Common Criteria compliant documents share a common and standardized terminology. It may be useful to read the glossary at the end of this document if some acronyms or terms are not familiar.

2 TOE Description

2.1 Introduction

The TOE implements a software system for unobservable and anonymous message delivery on an open network such as the Internet. More precisely, the system is implemented using a structure known as

MIX network, which, in brief, is a network of remailer systems, through which a message transits in an encrypted form, along a path (MIX chain) chosen by the sender user. The aim of the system is to hide the correspondence between origin and destination of a message to possible attackers and it is structured in such a way that not even the MIX nodes visited by the message know its full path.

2.2 TOE Structure

The TOE consists of all the software, hardware and firmware supporting the operation of the unobservable message transmission application. A synthetic overview of what software systems belong to this TOE, in comparison with the other two mentioned Protection Profiles ([Iac99b], [Iac99a]) is shown in table 1.

TOE (named by it's PP)	components		
Single MIX PP	single MIX		
PP for an Unobservable Message Delivery Application using MIXes	single MIX	MIX network	
User-Oriented PP for Unobservable Message Delivery using MIX networks	single MIX	MIX network	client software

Table 1: Components belonging to the TOEs

As may be seen, this TOE includes the whole system between the end-users (origin and destination), including the hosts and software running the single MIXes, the connection between the single MIXes, and the client software used by the remote users to access the MIX network services. Figure 1 shows such a TOE in a graphical form. The user sees the MIX network as a black box hidden behind the client software, which acts as interface to the system.

The TOE is composed by a set of independently administered MIX nodes, whose operation is described by [Cha81], which operate in a networked environment (which can be thought of as an open network like the Internet), and exchange and process batches of messages. These messages are in turn generated by the users, as the normal payload of the network, or by the MIX nodes as dummy messages, or control messages[1]. These nodes normally run under the technical supervision of an administrator, and are sponsored by an organization, which in turn may charge the users for the service.

The TOE consists also of the client software, normally not explicitly administered, which is used by the users to connect to the rest of the TOE (the MIX network) and send and receive messages.[2] This client software must implement both cryptographic and key management functionalities, and must be trusted by the user to allow a correct and safe operation of the MIX network, with respect to threats to the user, originating, in particular from the MIX network, with a multilateral view of security

The main subjects which interact or form the TOE are thus,

- the user, who originates and receives messages,

[1] For example, key expiration or renewal messages.

[2] A real world example of such client software may be the Freedom client (see [FRE]) or the MIXmaster software, used as client. See ([Cot]).

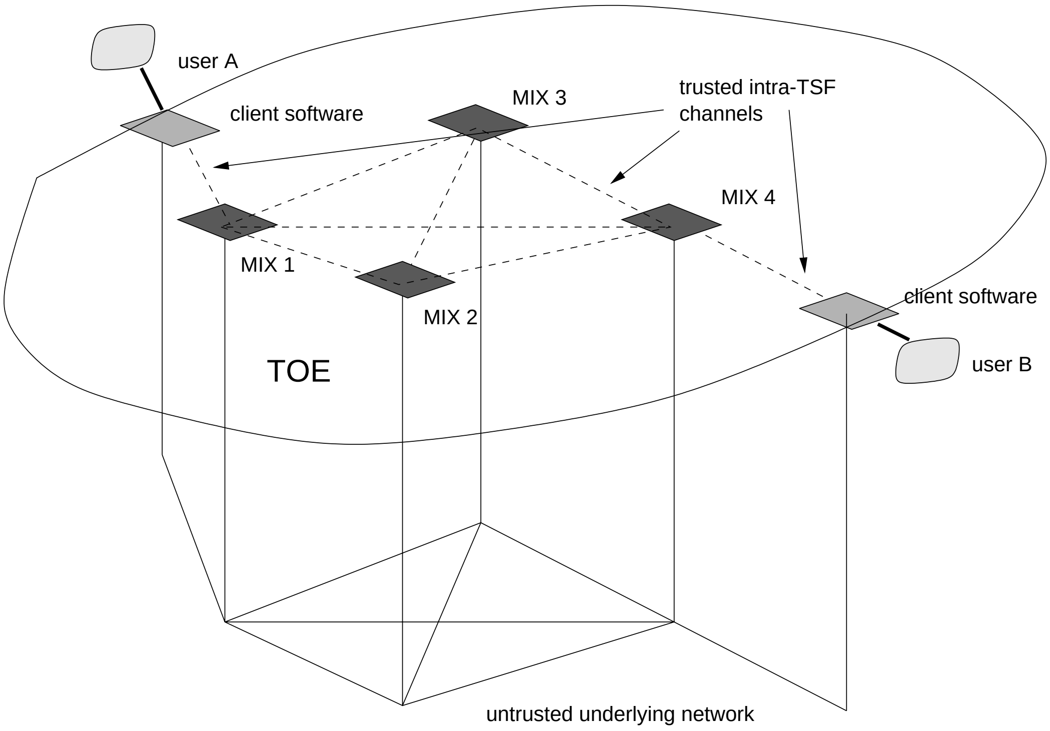

Figure 1: TOE from a user's perspective

- the client software,
- the various MIX nodes, which run a MIX application software as described in [Iac99b],
- the administrators (who can also become attackers),
- possibly, other attackers (who act in many different ways).

2.2.1 Assets

The assets involved in the operation of the TOE, and which require protection, are[3]:

- the user's message, which can be thought of as a block of text transmittable using the conventional e-mail delivery mechanisms,
- the auxiliary information related to the message:
  - its content,
  - its existence[4],
  - the delivery time and date,
  - the destination (intended as final destination user),
  - the origin (initial origin user),

[3] Note that this PP is concerned only about the assets which relate directly to the user.

  - the chosen to be traversed path,

- the user's cryptographic keys, and any related information,
- any other security related information.

2.2.2 Threat agents

The threat agents are attackers, which may be divided in two broad categories: *active* attackers, and *passive* attackers. The former actively interfere with the TOE operation, with the communication between the user and the TOE, while the latter only listen to the communications between parts of the TOE, the user and the TOE, or try to exploit covert channels to gain knowledge on the TOE's internal operation. In general an attacker is some subject interested either in a specific communication, or in tracing all communications, or in tampering with a part of the TOE, as a MIX node, client application software, and so on.

Another source of threats are *errors*, which might be processing errors, caused by bugs in the TOE software, or transmission errors, caused by other entities as the communication network, the underlying operating systems, etc.

2.2.3 Threats and attack strategies

Table 2 shows a synthetic overview of the threats that the TOE must address.

The first group of threats includes the main categories caused by active and passive attackers. Note however that this is not a comprehensive list of all the threats to which a MIX node or MIX network is subject to; instead, the threats here listed are those which more closely relate to the user of a MIX system. The threats directly aimed at the MIX network are not covered by this Protection Profile.

The second group collects some common error causes. This second group of threats is obviously to be taken care of not only to avoid the direct consequences (misdelivery, loss...), but also because an error in the application software implementing a MIX might expose an exploitable covert channel.

Keep also present that the list of threats is not fully orthogonal: this means that some of the threats actually partially overlap, (e. g. the content disclosure and traffic analysis); neither is the list complete. The overlap between threats derives from many possible causes, among which: overlaps in pre-existant threat lists, overlaps in attack implementation, similar goals, etc. The uncompleteness derives from the fact that this PP is concerned with the user perspective of the MIX network; a complete threat list for the single MIX *nodes* can be found in [Iac99b].

3 Security Environment

This section formally lists and describes the assumptions, threats and organizational policies for this TOE, and is thought to be compliant with version 15408-3 FDIS of the Common Criteria. [JTC98a]

[4] Statistical data may be gathered and used for traffic intensity analysis.

Name	Type	Implementation	Analysis	Effect	Countermeasure(s)
content disclosure	passive	messages are intercepted while transiting between parts of the TOE	of incoming and outgoing traffic	information on use patterns	generation of dummy messages by the users, traffic shaping
untrusted MIX	active	a MIX in the network may be compromised and hold, process and/or disclose tracing information	of transiting messages	exposure of linking information	division of trust
MIX conspiracy	active	some MIXes in the network may conspire to share and analyze traffic information	of transfer logs and TSF operation	loss of expected security	organizational policies:
forgery	active	an attacker may send forged messages using a user's origin credentials	n.a.	loss of accountability properties	use of digital signatures
network block	active	block the network connections to part of the TOE	n.a.	interruption of service degraded service	organizational policies and distribution of the TOE
traffic analysis	passive	intercept and read messages exchanged by parts of the TOE	message content	leak of information on message paths	encryption ease of use
redirection	active	intercept and redirect the network connections to part of the TOE	of requested traffic in the redirected network	loss of expected security	encryption
misuse	error	erroneous use of the TSF by the user	n.a.	loss of expected security	documentation
processing error	error	accidental processing error resulting in truncation, loss, alteration of messages	n.a.	unreliable service	redundancy assurance techniques
unreliability	error	the connecting network may be unreliable, resulting in message loss, truncation or alteration	n.a.	unreliable service	redundancy error detection

Table 2: Taxonomy of threats as seen by a MIX user

3.1 Assumptions

This section includes all relevant assumptions identified prior to or during the process of writing this PP. Assumptions are divided in four broad categories, i.e. assumptions about the intended use of the TOE, about logical and physical protection, about connectivity and, finally, about personnel concerns.

3.1.1 Assumptions about the intended use of the TOE

3.1.1.1 A.SecurityGoals *The TOE is assumed to be used to achieve unlinkable and anonymous or pseudonymous communication. Other security properties, as unobservability of TOE use are not contemplated.*

3.1.2 Physical/logical protection of the TOE

3.1.2.1 A.LogicalSec *The TOE will perform as long as the user take care of securing the logical access to their computing environment.*

Note: This assumption requires some explanatory text. As logically securing mainstream operating systems and environments, especially when networked, is close to impossible[5], the assumption should be taken rather loosely, provided that if the risk analysis leads to the conclusion that an attack on the user's workstation is likely, then the user should adopt a safer operating environment.

3.1.2.2 A.OS *The single parts of the TOE run on operating system platforms which are assumed to be trusted and not to expose privacy related information belonging to the TOE.*

3.1.2.3 A.PhysSec *The TOE will perform as long as the user take care of securing their physical access to the message traffic handled by the TOE.*

Note: This is a point which can not be over-stressed; an insecure physical user location may be easily exploited against the user who mistakenly believes that his or her communications are unobserved.

3.1.3 Connectivity aspects

3.1.3.1 A.MinimalConnectivity *The TOE might not be able to reach its goal if an attacker is able to block all access points of the user to the MIX network .*

3.1.3.2 A.MinimalTrust *The TOE might not be able to reach its goal if all nodes (MIXes) of the network are subverted.*

3.1.3.3 A.OpenEnvironment *The MIX network works in an open networked environment.*

[5] As shown, e. g., by the "back orifice", a software product which, once easily installed on a networked Windows 95 or 98 machine is able to monitor any system activity, including the user's activity on the graphical desktop user interface.

3.1.3.4 A.UnreliableNetwork *The connecting network might not be reliable on correctly delivering messages between parts of the TOE. Specifically, messages may be lost, altered or truncated accidentally.*

Note: the TOE is however not required to provide reliable service. A high degree of reliability may be achieved by sending multiple copies of a message through different paths.

3.1.4 Personnel aspects

3.1.4.1 A.UserCooperation *Users cooperate actively at the enforcement of the security policy of the TOE.*

Note: Users are trusted to use in a correct manner the services made available by the TOE to reach their anonymity goals.

3.2 Threats

A formal list of threat to the TOE is presented here, which, along with the operating assumptions above constitutes the formal description of the security-relevant aspects of the TOE.

3.2.1 Threats to be addressed by the TOE

This section lists threat which are to be countered by the TOE alone.

3.2.1.1 T.ContentDisclosure *An attacker might intercept transiting messages between parts of the TOE and read their content, thus disclosing it, together with any related information.*

Note: This is a threat not only to the operation of the TOE (as discussed in [Iac99b]), but also for the user, whose communications might be traced. In particular, this threat relates to messages transiting from the user client to a node on the network and refers to both the original message content (written by the user), and also to the routing information and other auxiliary information carried by the message.

3.2.1.2 T.EndPointTrafficAnalysis *An attacker might intercept transiting messages between parts of the TOE (user client and MIX node), and use the related information to perform traffic analysis on a user.*

Note: this threat relates to the concepts of *sender anonymity* and *receiver anonymity.* As viewed traditionally, main goal of the MIX network is to hide the relation between receiver and sender of a message (this property also known as *sender/receiver anonymity*). However, once a suspect on a possible communication between two users is established, it may be possible to monitor the end points of message chains for a statistical correlation between transmission and reception times, especially if the traffic on the network is low, the users few, and the per-user traffic low.[6]

[6] A similar discussion, related to Web transactions, may be found in [RR98].

3.2.1.3 T.KeyForgery *An attacker might generate forged keys, simulating the activity of a given MIX, distribute them, and make the user employ them to encrypt message in the belief that such messages are only readable by the replaced MIX.*

Note: This is a threat to the originating user, who will send messages readable to an attacker, and might not be warned about it. A trust scheme (implemented for example by a certification authority) is required to counter this threat.

3.2.1.4 T.Misuse *The user might install, configure or use the TOE interaction functions in an insecure manner, hence compromising the expected security properties offered by the TOE.*

Note: This threat is particularly relevant when considering the "human" element when this is the user, because the user is not expected to have as deep a knowledge about the TOE functions and about the security concerns as, for example, a system administrator, who represents the human element in the case of an administered MIX node.

3.2.1.5 T.OneStepPath *A MIX may gain information linking origin and destination if the path from the origin user to the destination user contains only one MIX.*

3.2.1.6 T.UntrustworthyMIX *Some MIX(es) in the network may be compromised and hold, process and/or disclose information useful to trace, and/or reveal the content of, communications.*

3.2.2 Threats to be addressed by the Environment

This section lists threats which are to be countered even only in part by the operating environment, i. e. the organizational and procedural frame, of the TOE. This does not mean that support in the TOE is not needed to counter the threats, but only that such support is not sufficient to counter them completely.

3.2.2.1 TE.MIXConspiracy *Some MIXes in the network may be compromised and share information useful to trace, and/or reveal the content of, communications.*

Note: This threat represents an extension to the **T.UntrustworthyMIX** threat, in that it introduces the concept of information sharing between parts of the TOE.

3.2.2.2 TE.PartialNetworkBlock *An attacker might block the connection between parts of the TOE and the user.*

Note: This is a typical DoS attack, where part or the entire TOE is rendered unusable.

3.2.2.3 TE.Redirection *An attacker might redirect the connections between parts of the TOE and act as to replace that part seamlessly, thus effectively acting as a compromised MIX subset.*

Note: See **T.UntrustworthyMIX** (3.2.1.6) above.

3.3 Organizational Security Policies

This section includes the two main policies which describe the functionality provided by the TOE to the user: i. e. untraceability and anonymity.

3.3.1.1 O.Anonymity *The TOE shall provide for an anonymous message delivery service; that is, the recipient of a message shall not be able to know the origin of the message, unless the author expressly inserts this information in the message body.*

3.3.1.2 O.Untraceability *The TOE shall provide for an untraceable message delivery service; this means that, taken any message transiting through the system at any time, it shall not be possible to obtain enough information to link its origin and destination users.*

4 Security Objectives

This section lists the security objectives for the TOE; the security objectives are formal statements which state the requirements specifications for the TOE (with regard to security issues). They are not intended to describe how the objectives should be met.

4.1 Security Objectives for the TOE

4.1.1.1 SO.AdequateDocumentation *The TOE shall provide the user with adequate, readable documentation on the correct use of the security functions.*

4.1.1.2 SO.Anonymity *The TOE shall accept and process messages without requiring that the processed data may be in any way linked to the origin user.*

See: 3.3.1.1.

4.1.1.3 SO.ConcealMessageContent *The TOE shall enforce that the content of all messages transiting on the network be unaccessible to all third parties, in whatever point of the network the messages are intercepted.*

4.1.1.4 SO.CounterTrafficAnalysis *The TOE shall be constructed as to counter traffic analysis techniques specifically aimed at analyzing the communications between user client software and the MIX network.*[7]

4.1.1.5 SO.DivideSecurityInformation *The TOE shall be constructed as to provide the user the ability, and enforce the correct use of such ability, of determining the allocation of unlinkability-relevant data among different parts of the TOE.*

[7] The concept of protecting also the first step of communications from the user to the network is described in length in [FRE].

4.1.1.6 SO.DivideSecurityProcessing *The TOE shall provide to the user the ability, and enforce the correct use of such ability, of freely choosing a combination of MIX nodes among which to allocate the processing activities achieving unlinkability.*

4.1.1.7 SO.EnforceProperUse *The TOE (and especially the user interface part of the TOE) shall enforce the proper and secure use of the security functions of the TOE.*

Note: for example, require secure pass phrases, encryption, minimum message chain length…

4.1.1.8 SO.EnforceTrustDistribution *The TOE shall be constructed to enforce the user's choice of information and processing distribution.*

4.1.1.9 SO.Identity *The TOE shall uniquely identify the single MIX nodes and users and provide means to transmit data to a specific MIX while preserving the confidentiality of such data.*

4.1.1.10 SO.KeyTrustAssurance *The TOE shall provide the user the ability, and enforce the correct use of such ability, of validating any public key used for encryption purposes against some trusted mechanism, to gain confidence that the communicating partner is actually who he claims to be.*

4.1.1.11 SO.MinimizeSecurityInformation *The TOE shall be constructed as to minimize the use, distribution and availability time frame of information impacting unlinkability.*

4.1.1.12 SO.Untraceability *The TOE shall also ensure that no subject (user, administrator, threat agent) has the possibility to gain sufficient information as to track back the origin of a message.*

See: 3.3.1.2

4.2 Security Objectives for the Environment

4.2.1.1 SOE.AntagonisticManagement *The TOE shall be independently and antagonistically managed.*

Note: the main problem with this security objective to be fulfilled by the environment, is that it is nearly impossible to enforce it without some form of post-deployment assurance evaluation control and maintenance.

4.2.1.2 SOE.DistributedNetwork *The TOE shall rely on a topologically distributed network.*

Note: this is required to maximize the resources which an attacker must deploy in the attempt to "cut off" part of the network from the rest. Apart from requiring specific design choices, this requirement can only be met by implementing a sound collective administration policy, and by providing means to assure the users of the effects of such a policy.

4.3 Security Objectives Rationale

Table 3 shows a cross-indexing of threats and organizational policies with security objectives. A check mark indicates that the security objective is needed to counter the respective threat.

	O.Anonymity	O.Untraceability	T.ContentDisclosure	T.EndPointTrafficAnalysis	T.KeyForgery	T.Misuse	T.OneStepPath	T.UntrustworthyMIX	TE.MIXConspiracy	TE.PartialNetworkBlock	TE.Redirection
SO.AdequateDocumentation						*					
SO.Anonymity	*										
SO.ConcealMessageContent			*								
SO.CounterTrafficAnalysis				*							
SO.DivideSecurityInformation							*	*			
SO.DivideSecurityProcessing							*	*			
SO.EnforceProperUse						*					
SO.EnforceTrustDistribution						*	*	*	*		
SO.Identity								*			*
SO.KeyTrustAssurance					*						
SO.MinimizeSecurityInformation								*			
SO.Untraceability		*									
SOE.AntagonisticManagement									*		
SOE.DistributedNetwork										*	*

Table 3: Security Objectives to Threats and Organizational Policies mapping

4.3.1 Threats and Policies to Objectives mapping

The following rationale shows how each threat is countered by one or more security objectives.

O.Anonymity This organizational policy is **fully** satisfied by the **SO.Anonymity** objective.

O.Untraceability This organizational policy is **fully** satisfied by the **SO.Unlinkability** objective.

T.ContentDisclosure This threat is **fully** countered by the **SO.ConcealMessageContent** objective. If the messages are encrypted using a public key algorithm, and the keys are properly administered and stored, no external attacker may gain access to the content of a message.

T.EndPointTrafficAnalysis This threat is **fully** countered by the **SO.CounterTrafficAnalysis** objective. This objective may be implemented using various techniques, which are however generally different from the techniques used by countering traffic analysis in MIX networks, because traffic to and from the users has a different statistical behavior as compared to the traffic generated by a MIX node.

T.KeyForgery This threat is **fully** countered by the **SO.KeyTrustAssurance** objective. This objective may be reached using various trust assurance and validation schemes, (i. e. CA, trusted introducer, ICE-TEL[8]. . .).

T.Misuse This threat is **fully**[9] countered by the **SO.AdequateDocumentation**, **SO.EnforceProperUse** and **SO.EnforceTrustDistribution** objectives. Namely, the first objective provides the user with relevant documentation about the use of the TOE security functions, the second enforces the verification of all user actions and choices to be correct and secure, while the third provides the user the ability (and requires the TOE to enforce its use) to choose a chain of MIX nodes in a secure fashion.

An assumption (**A.UserCooperation**, see 3.1.4.1) is finally made to cover user's behavior with regard with security concerns (i. e. that the user will not compromise the security functions by, for example, divulging private keys.)

T.OneStepPath This threat is countered by the **SO.EnforceTrustDistribution** and **SO.DivideSecurityProcessing** objectives. The latter gives the users the possibility to allocate the processing activities among multiple nodes in the network, and the former enforces the user's choices in such direction.

T.UntrustworthyMIX This threat is **fully** countered by the **SO.DivideSecurityInformation**, **SO.DivideSecurityProcessing**, **SO.EnforceTrustDistribution**, **SO.Identity** and **SO.MinimizeSecurityInformation** objectives. If a compromised MIX node has access to a limited amount of information to abuse of, the resulting loss in terms of security properties will be limited. If the compromised MIX node does not work in cooperation with other MIX nodes, and the user correctly employs

[8] A description of the ICE-TEL "Internetworking Public Key Certification Infrastructure for Europe" certification scheme may be found in [CYC97].

[9] To the extent to which a misuse of the TOE is preventable from the TOE itself.

the security features provided by the MIX network, no harm can derive by such node. The **SO.Identity** allows the addressing of information to individually selected nodes of the TOE.

TE.MIXConspiracy This threat is **partly** countered by the **SO.EnforceTrustDistribution** and **SOE.AntagonisticManagement** objectives. It is obviously impossible to gain 100% assurance that MIX administrators do not conspire against the users. The proposed objectives help in meeting this requirement, but are not sufficient. However, a high number of MIXes in the network, and a requirement stating that each (or at least many of them) should be involved in the processing of a message may help counter the threat of conspiracy, because of the difficulty of organising a conspiracy on a wide and consistent fashion. Actually, if we look at the TOE from a user's standpoint, the MIX requires not only that functional properties be stated and required to guarantee against conspiracy, but also that assurance measures be taken to verify and control the system during its operation.

The **A.MinimalTrust** (see 3.1.3.2) assumption states that the TOE will not fulfill its goal if the network is completely compromised.

TE.PartialNetworkBlock This threat is **partly** countered by the **SOE.DistributedNetwork** objective. A distributed topology makes is more difficult to block the connections between all the parts of the network. The size of the network plays also a role with regard to this threat.

The **A.MinimalConnectivity** (see 3.1.3.1) assumption is to be also considered with regard to this threat.

TE.Redirection This threat is **fully** countered by the **SOE.DistributedNetwork** and **SO.Identity** objectives. A distributed topology makes is more difficult to redirect part of the connections between parts of the TOE. Having each MIX node provide and use its own cryptographic key pair (and the user encrypt any plaintext addressed to such MIX node) makes it impossible for an attacker to read and thus "process" the message content.

4.3.2 Objectives to Threats and Policies mapping

This section shows that actually all security objectives are necessary to counter some threat or satisfy some policy.

SO.AdequateDocumentation This objective is necessary to partly counter the **T.Misuse** threat: adequate documentation is the only official source of information on the TSF for the user, and should document whatever procedures and configurations are necessary for the correct use of the TOE.

SO.Anonymity This objective is necessary to satisfy one of the two main policies for the TOE, that is, that it must provide for anonymous message delivery (**O.Anonymity**).

SO.ConcealMessageContent The objective is necessary to counter the **T.ContentDisclosure** threat.

SO.CounterTrafficAnalysis This objective is necessary to counter the **T.EndPointTrafficAnalysis** threat. The **SO.Untraceability** objective is not adequate for this purpose, because traffic analysis does not actually expose direct information on message tracing, but only reveals message exchange patterns between the user and the network; especially low traffic users are jeopardized by this kind of passive attack.

SO.DivideSecurityInformation This objective is necessary to counter the **T.OneStepPath** and **T.UntrustworthyMIX** threats.

SO.DivideSecurityProcessing This objective is necessary to counter the **T.OneStepPath** and **T.UntrustworthyMIX** threats.

SO.EnforceProperUse This objective is necessary to help counter the **T.Misuse** threat: besides of being informed on how the TOE operates (see **SO.AdequateDocumentation**) the user shall be also directed by the software itself to correctly operate the TOE.

SO.EnforceTrustDistribution This objective is necessary to counter the **T.OneStepPath**, **T.UntrustworthyMIX** and **TE.MIXConspiracy** threats. This objective is different from the SO.DivideSecurityInform and SO.DivideSecurityProcessing in that the latter provide the user the ability, and enforce the proper use of such ability by the user, to choose the way information and processing activities are distributed on the system, while the former enforces the TOE to respect such choices.

SO.Identity This objective is necessary to counter the **TE.Redirection** and **T.UntrustworthyMIX**; in the former case an attacker may want to simulate being a MIX, in the latter a MIX may want to access information meant for another node.

SO.KeyTrustAssurance This objective is necessary to counter the **T.KeyForgery** threat, and provides access to key certification mechanisms in the TOE.

SO.MinimizeSecurityInformation This objective is necessary to counter the **T.UntrustworthyMIX** threat. Minimization of information disclosure is an important strategy for handling with potentially untrusted subjects.

SO.Untraceability This objective is necessary to satisfy the **O.Untraceability** organizational policy.

SOE.AntagonisticManagement This objective, which is partially provided for by the environment, is necessary to help counter the **TE.MIXConspiracy** threat. Nobody, apart from the management structure of the network, may naturally assure that the nodes are independently administered, but the objective states one possible strategy for obtaining independence.

SOE.DistributedNetwork This objective, which is partially provided for by the environment, is necessary to help counter the **TE.Redirection** and **TE.PartialNetworkBlock** threats. The topology upon the TOE will be deployed is naturally outside the scope of this document, but this objective must nevertheless be stated.

5 TSF description

The TOE security functions[10] include all software, hardware and firmware directly or indirectly necessary to operate the TOE. The functional requirements described in this document, however, relate only to the security functions for the TOE as required by the user.

The TSF consist, therefore, of the following items:

- MIX node application software,
- software and hardware supporting the last item,
- user client software used to connect to the MIX network.

All communications between MIX nodes, and between MIX nodes and users are considered intra-TSF;[11] there are therefore no external communications channels, nor transfers outside TSF control.

The TSF Interface (TSFI) consists of two distinct interfaces. One is the user's interface to the client side software, which allows the user to send and receive messages mediated by the TOE. Such software is directly installed on the user's workstation, and is operated by the user on request, or activated by scheduling algorithms (for transmission delaying), or activated by incoming connections (when receiving messages). The second is the TSF management interface used by the administrators.[12]

The TSF Scope of Control (TSC) consists of all the interactions that can occur within the TOE and are subject to the rules of the TSP (described below). Such interactions include the following:

- user client software connects to a MIX node to send a message,
- MIX node connects to another MIX node to send a message, message batch (including possible dummy messages), control information (including key information),
- MIX node connects to client software to deliver a message,

[10] One of the questions raised during the development of this Protection Profile was if the TSF description should be inserted at this point to better understand the requirements on the Security Functions which follow below. One argument against the presence of this section was that the PP should be technology independent, as stated in [JTC98b, part 1], and the TSF description already gives a hint about the employed technology.

On the other side, a precise description of the boundary and interfaces of the TSF is helpful to the understanding of the security requirements, and of the general structure of the TOE. It is also true that while some notable example Protection Profiles such as [JW98], which describes a firewall system, lack a TSF description, the technology employed by a firewall is broadly known and well understood by the target audience of Protection Profiles, while MIX systems are not so well known by the public.

Hence, it was deemed useful to give at least a short, informative description of the TSF, so that the reader could easily understand the following chapters.

[11] Using the CC jargon: Internal TOE Transfer.

[12] This part of the TSFI is not, however, described in this document.

- MIX node processes a transiting message (this includes all operations described in [Iac99b]),
- MIX node generates, distributes, stores, accesses to, uses, destroys, cryptographic keys,
- MIX node generates dummy messages.

The sessions (interaction between the user and the TOE) are obviously conditioned by the operating environment on the user's side. As an informative description, it may help thinking at a session as a procedure similar at the interaction with a mail encryption program. To send messages, the user simply states the receiver and the MIX chain. No authentication, login, or accountability procedures are necessary, since the operation is anonymous. When receiving messages, the user may be informed in an asynchronous manner, or check personally for the availability of mail on some server, and then engage in an authentication procedure to decrypt the message content.

6 IT Security Requirements

The following sections contain the formal functional and assurance requirements for the TOE, taken from the Common Criteria, Version 15408-3 FDIS [JTC98a], and from [Iac99c]. Note that the rationale for the selected requirements is directly inserted in this document following the functional components catalog, and not given as separate section or document because it is thought that including directly the rationale after the requirement eases their comprehensibility, while not hindering the applicability of formal evaluation procedures on the document.

6.1 Functional Requirements

A list follows with a catalog of selected functional components drawn from the sources cited previously, and addressing the security needs for the TOE described in this PP. A summary of the selected components is presented in table 4.

Note that in the table, a star in the column marked with † indicates that the component is a new component from [Iac99c], while the ‡ column indicates that the component is a modified CC component in [Iac99c].

Various Security Function Policies are to be enforced by the TOE. Table 5 shows a short list of the SFPs labels, their purpose and the components they relate to.The policies will be described fully under the respective policy definition components, but it can be anticipated, to enhance the readability of the text, that the

- MUDAC (Mandatory MIX User Data Access Control) policy is devised to enforce that all data in the TOE shall be explicitly addressed to a subject, and that only that subject may be able to gain access to the data,
- the CCE (Covert Channel Elimination) policy is devised to allow for elimination of covert channels in the MIX, to counter traffic analysis attacks.

Functional components		†	‡
FCS_COP.1	Cryptographic operation		
FCS_CKM.1	Cryptographic key generation		
FCS_CKM.2	Cryptographic key distribution		
FCS_CKM.4	Cryptographic key destruction		
FDP_ACC.2	Complete access control (MUDAC)		
FDP_ACF.1	Security attribute based access control (MUDAC)		
FDP_IFC.1	Subset information flow control (CCE)		
FDP_IFF.4	Partial elimination of illicit information flows		
FDP_IRC.2	Full information retention control	*	
FDP_ITT.1	Basic internal transfer protection		
FDP_RIP.2	Full residual information protection		
FIA_ATD.1	User attribute definition		
FIA_UID.1	Timing of identification		
FMT_MSA.1	Management of security attributes		
FMT_MSA.2	Secure security attributes		
FMT_MSA.3	Static attribute initialisation		
FMT_SMR.1	Security roles		
FPR_ANO.2	Anonymity without soliciting information		
FPR_TRD.2	Allocation of information assets	*	
FPR_TRD.3	Allocation of processing activities	*	
FPR_UNL.2	Unlinkability of users		*

Table 4: Summary of Functional Requirements

Label	Purpose	Related Components
MUDAC	Enforce access control on user data	**FDP_ACF.1, FDP_ITT.1**
CCE	Eliminate Covert Channels from user/MIX	**FDP_IFF.4**

Table 5: SFP enforced by the TOE.

6.1.1 Functional Requirements List

FCS_COP.1 Cryptographic operation

- **FCS_COP.1.1** The TSF shall perform [assignment: *key generation, decryption and encryption of messages, signature generation and verification*] in accordance with a specified cryptographic algorithm [assignment: *any adequate symmetric algorithm, using a suitable randomly generated key, transmitted using an asymmetric encryption algorithm*[13]] and cryptographic key sizes: [assignment: *any*] that meet the following: [assignment: *sufficient for the purpose of the MIX*].

  Application Note: the type and strength of the cryptographic functions must be specified by the ST, in accordance to the intended use of the TOE, and the perceived threats. See the **AVA_SOF** assurance security requirement below (see Paragraph 6.2.2.23.)

FCS_CKM.1 Cryptographic key generation

- **FCS_CKM.1.1** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *any suitable for the algorithm chosen in* ***FCS_COP.1.1***] and specified cryptographic key sizes [assignment: *any*] that meet the following: [assignment:

  - *sufficient key length for the purpose of the MIX*,
  - *each subject must generate its own keypair independently from the others (this applies also to MIX nodes)*

  ].

  Application Note: keys must be generated with an expiration date. The key lifetime and size must be chosen by the MIX administrator in function of the perceived threats of key leakage and brute force attack. The higher the threat, the shorter the key lifetime, and the greater the key size. See the **AVA_SOF** assurance security requirement below (see Paragraph 6.2.2.23.)

FCS_CKM.2 Cryptographic key distribution

- **FCS_CKM.2.1** The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [assignment: *certification by a CA and publication on multiple independent network sites*] that meets the following: [assignment: *standard to be specified by the ST*].

  Application Note: mandatory requirements for certification of cryptographic keys by a CA (Certification Authority) are here included without specifying how this should be implemented. This is because there are many different methods of certification and the choice depends on considerations over the environment, the intended use of the TOE, and organizational practice.

[13] Examples include using PGP or S/MIME cryptographic tools.

FCS_CKM.4 Cryptographic key destruction

- **FCS_CKM.4.1** The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: *remove the keys from the used key pool, issue a key revocation certificate*] that meets the following: [assignment: *standard to be specified by the ST*].

  Application Note: the precise procedures for key revocation and destruction are beyond the scope of this document; it is left to the ST to specify these procedures.

FDP_ACC.2 Complete access control (MUDAC)

- **FDP_ACC.2.1** The TSF shall enforce the [assignment: *Mandatory MIX User Data Access Control Policy*] on [assignment: *all subjects covered by the SFP, namely*

  - *MIX nodes,*
  - *client software,*
  - *users,*
  - *administrators,*

  and all objects generated by users, namely

  - *message content and*
  - *message routing information*

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

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

  Application Note: the Mandatory MIX User Data Access Control Policy (MUDAC for short) is stated as follows:

  1. all data produced by subjects covered by the SFP must obey the policy's requirements;
  2. data produced by subjects covered by the SFP must be explicitly addressed to some subject;
  3. data explicitly addressed to some subject must be unreadable by all other subjects;
  4. data produced by a subject may be read by the same subject;

  With regard to subjects which operate on behalf or on MIX nodes, this policy regards them as belonging to the administrative domain of the specific MIX, as defined in the **FPR_TRD.2** and **FPR_TRD.3** components.

FDP_ACF.1 Security attribute based access control (MUDAC)

- **FDP_ACF.1.1** The TSF shall enforce the [assignment: *MUDAC*] to objects based on [assignment: *message routing information, cryptographic key information*].
- **FDP_ACF.1.2** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment:

  - *a controlled subject is granted access to information explicitly addressed to it,*
  - *and to objects originated from it,*

  ].
- **FDP_ACF.1.3** The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment: *none*].
- **FDP_ACF.1.4** The TSF shall explicitly deny access of subjects to objects based on the [assignment: *none*].

FDP_IFC.1 Subset information flow control (CCE)

- **FDP_IFC.1.1** The TSF shall enforce the [assignment: *Covert Channel Elimination Policy*] on [assignment: *MIX nodes, client user software, messages, message transmission and reception*].

  Application Note: the Covert Channel Elimination (CCE) SFP requires the TOE to deploy techniques to eliminate covert channel exploitability by which an attacker may gain information about the use of the system by some user, especially with regards to traffic analysis information. The specific technique to adopt is not specified in this document. A discussion of traffic analysis viewed as an illicit information flow may be found in [Iac99b].

FDP_IFF.4 Partial elimination of illicit information flows

- **FDP_IFF.4.1** The TSF shall enforce the [assignment: *CCE*] to limit the capacity of [assignment: *information flow on user's use of the system through traffic analysis at the end nodes*] to a [assignment: *maximum capacity to be determined by the ST*].
- **FDP_IFF.4.2** The TSF shall prevent [assignment: *information flow on the systems' use patterns through traffic analysis at the end nodes*].

  Application Note: this functional component calls for partial elimination of illicit information flows, and is similar to part of the covert channel elimination requirements found in [Iac99b]; it relates however, not to the communications between MIXes, but to the communications between MIX and user (client software).

FDP_IRC.2 Full information retention control

- **FDP_IRC.2.1** The TSF shall ensure that all objects required for [assignment: *all activities undertaken by the TOE during its normal operation*] shall be eliminated immediately from the TOE upon termination of the activities for which they are required.

  Application Note: the only information which is not subject to information retention control is information needed to detect and prevent flooding attacks (see [Iac99b]); also this kind of information should be however studied to ensure that it is enough for detecting attacks, but that it is not useful for tracing message paths.

FDP_ITT.1 Basic internal transfer protection

- **FDP_ITT.1.1** The TSF shall enforce the [assignment: *MUDAC SFP*] to prevent the [selection: *disclosure, modification*] of user data when it is transmitted between physically-separated parts of the TOE.

FDP_RIP.2 Full residual information protection

- **FDP_RIP.2.1** The TSF shall ensure that any previous information content of a resource is made unavailable upon the [deallocation of the resource from] all objects.

FIA_ATD.1 User attribute definition

- **FIA_ATD.1.1** The TSF shall maintain the following list of security attributes belonging to individual users: [assignment:
  - *public cryptographic keys,*
  - *private cryptographic keys*].

FIA_UID.1 Timing of identification

- **FIA_UID.1.1** The TSF shall allow [assignment: *none*] on behalf of the user to be performed before the user is identified.
- **FIA_UID.1.2** The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

  Application Note: see 3.1.2.1; not all user sites allow for secure user identification; for this reason the requirement on user identification may or not apply to a specific TOE, and must be judged by a specific risk analysis.

FMT_MSA.1 Management of security attributes

- **FMT_MSA.1.1** The TSF shall enforce the [assignment: *MUDAC SFP*[14]] to restrict the ability to [selection: *change_ default, query, modify, delete*, [assignment: *none*]] the security attributes [assignment:

  – *selected route,*

  – *minimum number of hops to perform,*

  – *route randomization,*

  – *number of redundant messages to send,*

  – *traffic shaping parameters,*

  – *minimum encryption capabilities selection,*

  – *other security attributes*

  ] to [assignment: *the user originator of the data to which the security attributes refer*].

FMT_MSA.2 Secure security attributes

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

FMT_MSA.3 Static attribute initialisation

- **FMT_MSA.3.1** The TSF shall enforce the [assignment: *MUDAC SFP*] to provide [selection: *restrictive*] default values for security attributes that are used to enforce the SFP.
- **FMT_MSA.3.2** The TSF shall allow the [assignment: *user*] to specify alternative initial values to override the default values when an object or information is created.

  Application Note: this component is primarily targeted at the initialization of attributes by the user (route selection, minimum number of hops, etc. – see **FMT_MSA.1** above). Note also that only the user who created an object (a message) has the ability to operate on the objects security attributes.

FMT_SMR.1 Security roles

- **FMT_SMR.1.1** The TSF shall maintain the roles [assignment: *user, administrator*].
- **FMT_SMR.1.2** The TSF shall be able to associate users with roles.

[14] Point 4 of the policy.

FPR_ANO.2 Anonymity without soliciting information

- **FPR_ANO.2.1** The TSF shall ensure that [assignment: *all users covered by the SFP, the MIX administrators, the MIX node, threat agents*] are unable to determine the real user name bound to: [assignment:
  - *messages transiting through and processed by the MIX network*].
- **FPR_ANO.2.2** The TSF shall provide [assignment: *untraceable and anonymous message forwarding*] to [assignment: *all users under the TSF*] without soliciting any reference to the real user name.

FPR_TRD.2 Allocation of information assets

- **FPR_TRD.2.1** The TOE shall be divided in separate, independent, intercommunicating parts (administrative domains) governed by distinct access control and authentication configurations.
- **FPR_TRD.2.2** The distinct administrative domains of the TOE shall explicitly request access to data stored on other parts of the TOE to be granted access to it.
- **FPR_TRD.2.3** The TSF shall ensure that [assignment: *unlinkability related information, namely,*
  - *information produced by the transit of a message,*
  - *routing information,*
  - *timing information,*
  - *and all other relevant information*

  ] shall be stored [selection: *in a form unreadable by a single administrative domain of the TOE*] as to maintain the following conditions: [assignment: *no subject in the TOE, and no attacker, may gain enough information to fully trace a message chain from sender to receiver*].

  Application Note: the administrative domains correspond with the MIX nodes. This access policy provides an access point in the PP for the MIX "rotating" message encapsulation mechanisms, described in [Cha81].

FPR_TRD.3 Allocation of processing activities

- **FPR_TRD.3.1** The TOE shall be divided in separate, independent, intercommunicating parts (administrative domains) governed by distinct access control and authentication configurations.
- **FPR_TRD.3.2** The distinct administrative domains of the TOE shall explicitly request access to data stored on other parts of the TOE to be granted access to it.
- **FPR_TRD.3.3** The TSF shall ensure that [assignment: *single forwarding steps of messages*] shall be peformed by different administrative domains of the TOE, so that the following conditions are maintained: [assignment: *no subject in the TOE, and no attacker, may gain enough information to fully trace a message chain from sender to receiver*].

  Application Note: the administrative domains correspond with the MIX nodes.

FPR_UNL.2 Unlinkability of users

- **FPR_UNL.2.1** The TSF shall ensure that [assignment: *no user or subject*] are unable to determine whether [assignment: *users of the TOE sending messages through the TOE*] [selection: *are referenced by the same operation,* and *are referenced by the same object*].

  Application Notes: the operation here is considered to be that of processing a message by the TOE. The object is a message data item transiting in the TOE. A side-effect of this requirement is that not even the receiving user will be able to trace the message back to the sender, unless the sender explicitly includes evidence of its own identity the message.

6.1.2 Functional Requirements Rationale

This section provides a two-way rationale explaining both why each functional requirement is necessary and helps meet the objectives described earlier (Section 6.1.3), and how each security objective is partially or completely covered by the functional requirement components (Section 6.1.4). Table 6 shows a synthetic view of this relation. The stars indicate that the component is directly necessary to fulfill the related objective; plus-signs indicate that the component is required as dependency of some other component.

6.1.3 Functional requirements to security objectives mapping

FCS_COP.1 Cryptographic operation

This requirement is necessary to protect the confidentiality and integrity and accountability of user data, as required by the **SO.ConcealMessageContent** objective.

FCS_CKM.1 Cryptographic key generation

This requirement is necessary to ensure that cryptographic keys are generated following sound security practices. It indirectly supports the **SO.Identity** objective.

FCS_CKM.2, FCS_CKM.4 Cryptographic key distribution, Cryptographic key destruction

These requirements are necessary to ensure that cryptographic keys may be trusted by users to be the keys of the stated owners, as requested by the **SO.KeyTrustAssurance** objective.

FDP_ACC.2 FDP_ACF.1 Complete access control (MUDAC), Security attribute based access control (MUDAC)

These requirements name and define the scope of control of the MUDAC SFP, which is required by the **SO.EnforceTrustDistribution** objective, as well by the **SO.Identity** objective.

FDP_IFC.1 FDP_IFF.4 Subset information flow control (CCE), Partial elimination of illicit information flows

This requirements name and define the scope of control of the CCE SFP, which states the requirements for the elimination of covert channels in the TOE, as requested by the **SO.CounterTrafficAnalysis** objective.

	SO.AdequateDocumentation	SO.Anonymity	SO.ConcealMessageContent	SO.CounterTrafficAnalysis	SO.DivideSecurityInformation	SO.DivideSecurityProcessing	SO.EnforceProperUse	SO.EnforceTrustDistribution	SO.Identity	SO.KeyTrustAssurance	SO.MinimizeSecurityInformation	SO.Untraceability	SOE.AntagonisticManagement	SOE.DistributedNetwork
FCS_COP.1			*											
FCS_CKM.1									*					
FCS_CKM.2										*				
FCS_CKM.4										*				
FDP_ACC.2								*	*					
FDP_ACF.1								*	*					
FDP_IFC.1				*										
FDP_IFF.4				*										
FDP_IRC.2											*			
FDP_ITT.1			*											
FDP_RIP.2											*			
FIA_ATD.1									*					
FIA_UID.1					+	+								
FMT_MSA.1					*	*								
FMT_MSA.2					*	*	*							
FMT_MSA.3					*	*	*							
FMT_SMR.1					+	+								
FPR_ANO.2		*												
FPR_TRD.2					*			*					*	
FPR_TRD.3						*		*					*	
FPR_UNL.2												*		

Table 6: Functional Requirements to Security Objectives mapping

FDP_IRC.2 Full information retention control

This functional component requires the TOE to dispose of all information as soon as it is not more specifically required by the TOE. It helps meeting the **SO.MinimizeSecurityInformation** objective.

FDP_ITT.1 Basic internal transfer protection

This functional component states the TOE's requirements regarding the protection of the communications between parts of the TOE, with regard to confidentiality and integrity. It helps in meeting the **SO.ConcealMessageContent** objective.

FDP_RIP.2 Full residual information protection

This functional component requires the TOE to actively flush all non used information, both as a assurance measure for the user, with regard to untrusted MIXes, both to comply to the "minimization of deployed information" policy (**SO.MinimizeSecurityInformation**).

FIA_ATD.1 User attribute definition

This functional component is required to define the user attributes that the TOE shall maintain, in support to the **SO.Identity** objective.

FIA_UID.1 Timing of identification

This functional component is indirectly required by the **FMT_SMR.1** component, to establish a role authentication mechanism, by which the TOE may be administered.

FMT_MSA.1 Management of security attributes

The management of security attributes component is important to grant the user the right and possibility to effectively enforce the trust division policies (**SO.DivideSecurityInformation** and **SO.DivideSecurityProcessing**).

FMT_MSA.2, FMT_MSA.3 Secure security attributes, Static attribute initialisation

This component is needed by the attribute-based access control policy (MUDAC), to impose conditions on the TOE regarding the secure generation of default security attributes, and the secure state of user-defined security attributes. This helps in meeting the **SO.DivideSecurityInformation** and **SO.DivideSecurityProcessing** objectives.

Moreover both components are needed to satisfy the **SO.EnforceProperUse** objective, by requiring the TOE to provide secure default values and verify the security of inputted values of security attributes by the user.

FMT_SMR.1 Security roles

This component is needed by the security attribute management components, to define security roles for the TOE.

FPR_ANO.2 This functional component make sure that the TOE does not request identification information regarding the origin and destination of messages it handles, and that nobody may gain information linking a data object (message) to users. It aides in meeting the **SO.Anonymity** objective.

FPR_TRD.2, FPR_TRD.3 Allocation of information assets, Allocation of processing activities

These components are needed to specify requirements on the TOE to allow users to allocate processing activities and data assets on different parts of the TOE, as to support respectively the **SO.-DivideSecurityInformation** and **SO.DivideSecurityProcessing** requirements. Both components help meeting the **SO.EnforceTrustDistribution** objective. Moreover both components partly help also in meeting the **SOE.AntagonisticManagament** objective.

FPR_UNL.2 Unlinkability of users

This functional requirements make sure that the message processing algorithm does not implicitly expose information regarding the origin and destination of messages. It aides in meeting the **SO.Untraceability** objective.

6.1.4 Security objectives to functional requirements mapping

SO.AdequateDocumentation This objective is **not** covered by any functional component, but specific assurance components (**AGD_ADM.1**, **AGD_USR.1**) provide a similar set of requirements.

SO.Anonymity This objective is **fully** met by the **FPR_ANO.2** “Anonymity without soliciting information” functional component. This security objective requests that the user’s interaction with the TOE be completely anonymous. When a message is sent, the TOE must not retain sufficient information to track the message back to the user.

SO.ConcealMessageContent This objective is **fully** met by a number of different functional components, namely, **FCS_COP.1**, **FDP_ITT.1**. The first component requires the TOE to provide cryptographic operation security functions, while the **FDP_ITT.1** provides internal transfer protection.

SO.CounterTrafficAnalysis This objective is **fully**[15] met by the **FDP_IFC.1** and **FDP_IFF.4** functional components. The former names and defines the TSC of the CCE information flow control policy. The latter specifies requirements on the analysis and elimination of covert channels regarding traffic analysis on the connections between users and MIX nodes.

SO.DivideSecurityInformation This objective is **fully** met by the **FMT_MSA.1**, **FMT_MSA.2**, **FMT_MSA.3**, and **FPR_TRD.2** functional components, as shown in table 7. The security roles and security attribute management components define the user as a security attribute manager, and allow him or her to manipulate a number of security attributes regarding the user data that he or she generated. The trust distribution component specifies that security relevant information shall be divided

[15] Note that a full covert channel analysis may not be always viable, nor may it be possible to discover and eliminate all covert channels.

between parts of the TOE. The **FMT_MSA.2**, and **FMT_MSA.3** components, furthermore, define requirements on the default and user-defined values for security attributes.

Security Objective	Fulfilling Component(s)
"The TOE shall be constructed as to provide the user the ability,	**FMT_MSA.1 FMT_SMR.1**
and enforce the correct use of such ability,	**FDP_MSA.2 FMT_MSA.3**
of determining	**FPR_TRD.2**
the allocation of unlinkability-relevant data among different parts of the TOE."	**FPR_TRD.2**

Table 7: **SO.DivideSecurityInformation** fulfillment table.

SO.DivideSecurityProcessing This objective is **fully** met by the **FPR_TRD.3**, which specifies that the TOE shall be constructed in such a way to allow the execution of security relevant processing in a distributed fashion, as to protect the unlinkability-related processing activities; by the **FMT_MSA.2** and **FMT_MSA.3** components, which require the TOE to check the security of default and user defined security attribute settings. The **FMT_MSA.1** component allows the user to define security attributes for the data he or she originates (routing information), as required by the sentence: *"The TOE shall provide the user the with ability [. . .] of freely choosing a combination of MIX nodes*[16] *among witch to allocate the processing activities. . . "*).

SO.EnforceProperUse This objective is **fully** met by the **FMT_MSA.2** and **FMT_MSA.3** components, which require the TOE to check the security of default and user defined security attribute settings.

SO.EnforceTrustDistribution This objective is **fully** met by the **FDP_ACF.1**, **FDP_ACC.2**, and **FPR_TRD.2**, **FPR_TRD.3** functional components. The latter two require the TOE to enforce "trust distribution" mechanisms in order to avoid that subjects may gain enough information for tracing messages, and the protected information assets derive both from processing activities and from objects stored on the TOE. The user data access policy are to be respected by all subjects (including MIX nodes); according to such policy subjects may gain access to information only if this is explicitly addressed to them. This policy is in some way complementary to the trust distribution components, strenghtening the security properties of the TOE, and allowing for explicit controlled access by subjects to objects (for example for defining security attributes).

SO.Identity The objective is **fully** met by the **FIA_ATD.1**, **FDP_ACF.1**, **FDP_ACC.2**, **FCS_CKM.1** functional components. The first component defines the security attributes for the users. The next two components require the TOE to allow individual addressing of messages to MIXes and users, the **FCS_CKM.1** provides requirements for the key generation framework, namely, that keys are to be uniquely generated by the TOE.

[16] This is equivalent, in the CC jargon, to: "defining values of security attributes".

SO.KeyTrustAssurance This objective is **fully** met by the **FCS_CKM.2** "Cryptographic key distribution" and **FCS_CKM.4** ("Cryptographic key destruction") component, which requires the TOE to provide key distribution and certification capabilities.

SO.MinimizeSecurityInformation This objective is **fully** met by the **FDP_RIP.2** and **FDP_-IRC.2** functional components. The first component requires the TOE to eliminate any trace of information when deallocating such information. The second and third components require the TOE to gain access to the minimum amount of user information as strictly required for the correct operation of the system. The minimum time constraint is not covered by any component.

SO.Untraceability This objective is **fully** met by the **FPR_UNL.2** "Unlinkability of users" functional component. The TOE must ensure that threat agents are not able to collect sufficient data as to link message origin and destination.

SOE.AntagonisticManagement This objective is **partly** met by the **FPR_TRD.2** "Allocation of information assets" and **FPR_TRD.3** "Allocation of processing activities" components. However functional interoperability and capability of separate administration and independent operation is as much as the TOE PP can require; *political and economical issues of this kind must be assessed after deployment of the TOE.*

SOE.DistributedNetwork This objective is **not** covered by any functional component.

The topological structure must be adequate to support the MIX network in such a way that compromission of all communication links between MIX nodes and users is unfeasable.

However, it would be outside of the scope of this document to state a more in-depth requirement.

6.2 Assurance Requirements

This section describes the assurance requirements of the TOE, and is written following the specifications of Version 15408-3 FDIS of the CC [JTC98a]. The assurance requirements are taken from part 3 of the Common Criteria, and the selected EAL (Evaluation Assurance Level) for this TOE is EAL 5.[17] This level is chosen to gain a high level of assurance that the TOE will be developed, delivered, and evaluated following rigorous commercial practices. A formal model of the TOE security policies must be provided and evaluated[18], and the system must be independently tested. It must be stressed that the selected EAL for a specific TOE is also dependent from a case-by-case risk analysis of the intended use and environment of the TOE.

EAL 5 is necessary to satisfy these requirements (independent testing and formal model specification and evaluation), because it is the first level in which the **AVA_CCA.1** "Covert Channel Analysis" and the

[17] Note however that while this document requires EAL 5, the actual choice of an EAL is both largely independent from the statement of the functional requirements and may also be eventually modified.

This choice, and actually the whole Assurance Requirements section, should as such be taken as an *example* of how an EAL is chosen and justified, and does not necessarily represent the best or most viable choice of an EAL for a specific application.

In fact, while the justification for the selection of EAL 5 in this section is correct, such a high level may impose an excessive burden on the development process for this kind of application.

[18] Formal models of the mechanisms employed in MIX systems are available in the literature.

ADV_SPM.3 "Formal TOE security policy model" assurance requirements appear, which are required to perform an in-depth analysis of the security properties of the software applications implementing the MIX nodes of the network.

This EAL is also deemed sufficient, considering the high cost of adopting the next level of assurance (EAL 6), with respect to the moderate benefits deriving from it. As said above, the vulnerability assessment must be made on the base of the intended threats and use of the system. Furthermore, the advanced development practices prescribed by EAL 6 are not necessary for a simple TOE like the one described in this PP; finally, the life cycle assurance requirements introduced by EAL 6 are considered not particularly useful for this kind of system.

6.2.1 Assurance requirements rationale

Table 8 shows the assurance components required by EAL 5 (augmentations are shown in boldface characters).

Assurance Class	Assurance Components
Configuration management	ACM_AUT.1 Partial CM automation ACM_CAP.4 Generation support and acceptance procedures ACM_SCP.3 Development tools CM coverage
Delivery and Operation	**ADO_DEL.3 Prevention of modification** **ADO_IGS.2 Generation log**
Development	ADV_FSP.3 Semiformal functional specification ADV_HLD.3 Semiformal high-level design ADV_IMP.2 Implementation of the TSF ADV_INT.1 Modularity ADV_LLD.1 Descriptive low-level design ADV_RCR.2 Semiformal correspondence demonstration ADV_SPM.3 Formal TOE security policy model
Guidance documents	AGD_ADM.1 Administrator guidance AGD_USR.1 User guidance
Life cycle support	ALC_DVS.1 Identification of security measures ALC_LCD.2 Standardised life-cycle model ALD_TAT.2 Compliance with implementation standards
Tests	ATE_COV.2 Analysis of coverage ATE_DPT.2 Testing: low-level design ATE_FUN.1 Functional testing **ATE_IND.3 Independent testing - complete**
Vulnerability assessment	AVA_CCA.1 Covert channel analysis AVA_MSU.2 Validation on analysis AVA_SOF.1 Strength of TOE security function evaluation AVA_VLA.3 Moderately resistant

Table 8: Assurance Requirements: EAL.5 (augmented)

Although EAL 5 was selected as a general assurance requirements framework for this TOE, it was deemed however necessary to include also some higher level components to the selected EAL, namely, the **ADO_DEL.3** “Prevention of modification”, **ADO_IGS.2** “Generation log”, and **ATE_IND.3** “Independent testing - complete” components. A rationale for these augmentations is given below.

- **ADO_DEL.3**: this assurance component augments the standard **ADO_DEL.2** by requiring the TOE’s documentation to explain how to enforce the detection of modification of the TOE.
- **ADO_IGS.2**: this component augments **ADO_IGS.1** and requires the documentation to describe procedures to gain knowledge (via a log) on how and when the TOE was generated.

  The previous two components are relevant in that parts of the TOE might be compromised by an attacker simply by substitution of some software component, with a new one containing a trojan horse or some other device, or by alteration of the configuration parameters of the TOE, and give the attacker access to information about the usage and message exchange patterns of the TOE.
- **ATE_IND.3**: this component augments **ADO_IND.2** and requires the evaluator to independently and systematically apply all tests on the TOE to confirm the developer tests results.

  The previous component is relevant in that an independent test of the TOE is necessary to gain confidence that the TOE behaves as stated by the developer and expected by the user. All parts of the TOE must be tested.[19]

6.2.2 Formal assurance requirements

The list of formal assurance requirements shown in table 8, drawn from Version 15408-3 FDIS of the CC follows.

6.2.2.1 ACM_AUT.1 Partial CM automation

Developer action elements:

- ACM_AUT.1.1D The developer shall use a CM system.
- ACM_AUT.1.2D The developer shall provide a CM plan.

[19] What is here overlooked, but must be kept present while deploying this kind of TOE, is that a one-time evaluation is not sufficient to give the user assurance that the TOE will perform as intended and expected. This is because a compliant TOE might be modified *after* being deployed, and thus fail delivering the expected security properties, while the user is maliciously induced into thinking that using of the TOE is safe. A kind of security property assurance evaluation maintenance program is thus required, to monitor constantly and independently the TOE, during its operation.

On the other side, it is clearly stated in the CC [JTC98a, part 3, page 138], that the post-deployment security assurance cannot be determined at the time of the TOE evaluation, and no assurance component extends its effect beyond the delivery of the TOE – particularly to the time of operation and use.

This leads to an interesting conclusion regarding the adequacy of the Common Criteria for describing and grading Information Technology products with regard to security; that is, the CC allow only for *static* evaluation, and do not specify any procedures for the maintenance of such evaluation results. A continuative evaluation process, however, is *essential* for some kinds of security applications, in particular, where the user *does not* trust the deployer (administrator, sponsor…) to maintain the security properties of the product. To use a real world example, the bank system, while largely trusted by the users (the clients), is nevertheless constantly controlled by the authorities to spot any infringements of the user’s rights and state regulations. Why should a software product not be granted a similar evaluation scheme, also considered that the trust factor in this case is missing?

In conclusion, it is deemed necessary at this point state that a product such as the described TOE cannot be considered fully evaluated and *secure* if the assurance requirements provided by the CC are not augmented by some other requirements which allow for a continuative evaluation scheme.

Content and presentation of evidence elements:

- ACM_AUT.1.1C The CM system shall provide an automated means by which only authorised changes are made to the TOE implementation representation.
- ACM_AUT.1.2C The CM system shall provide an automated means to support the generation of the TOE.
- ACM_AUT.1.3C The CM plan shall describe the automated tools used in the CM system.
- ACM_AUT.1.4C The CM plan shall describe how the automated tools are used in the CM system.

Evaluator action elements:

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

6.2.2.2 ACM_CAP.4 Generation support and acceptance procedures

Developer action elements:

- ACM_CAP.4.1D The developer shall provide a reference for the TOE.
- ACM_CAP.4.2D The developer shall use a CM system.
- ACM_CAP.4.3D The developer shall provide CM documentation.

Content and presentation of evidence elements:

- ACM_CAP.4.1C The reference for the TOE shall be unique to each version of the TOE.
- ACM_CAP.4.2C The TOE shall be labelled with its reference.
- ACM_CAP.4.3C The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.
- ACM_CAP.4.4C The configuration list shall describe the configuration items that comprise the TOE.
- ACM_CAP.4.5C The CM documentation shall describe the method used to uniquely identify the configuration items.
- ACM_CAP.4.6C The CM system shall uniquely identify all configuration items.
- ACM_CAP.4.7C The CM plan shall describe how the CM system is used.
- ACM_CAP.4.8C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

- ACM_CAP.4.9C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.
- ACM_CAP.4.10C The CM system shall provide measures such that only authorised changes are made to the configuration items.
- ACM_CAP.4.11C The CM system shall support the generation of the TOE.
- ACM_CAP.4.12C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

Evaluator action elements:

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

6.2.2.3 ACM_SCP.3 Development tools CM coverage

Developer action elements:

- ACM_SCP.3.1D The developer shall provide CM documentation.

Content and presentation of evidence elements:

- ACM_SCP.3.1C 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, security flaws, and development tools and related information.
- ACM_SCP.3.2C The CM documentation shall describe how configuration items are tracked by the CM system.

Evaluator action elements:

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

6.2.2.4 ADO_DEL.3 Prevention of modification

Developer action elements:

- ADO_DEL.3.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.
- ADO_DEL.3.2D The developer shall use the delivery procedures.

Content and presentation of evidence elements:

- ADO_DEL.3.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.
- ADO_DEL.3.2C The delivery documentation shall describe how the various procedures and technical measures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.
- ADO_DEL.3.3C 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:

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

6.2.2.5 ADO_IGS.2 Generation log

Developer action elements:

- ADO_IGS.2.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

- ADO_IGS.2.1C The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.
- ADO_IGS.2.2C The documentation shall describe procedures capable of creating a log containing the generation options used to generate the TOE in such a way that it is possible to determine exactly how and when the TOE was generated.

Evaluator action elements:

- ADO_IGS.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- ADO_IGS.2.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

6.2.2.6 ADV_FSP.3 Semiformal functional specification

Developer action elements:

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

Content and presentation of evidence elements:

- 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.
- ADV_FSP.3.2C The functional specification shall be internally consistent.
- 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.
- ADV_FSP.3.4C The functional specification shall completely represent the TSF.
- ADV_FSP.3.5C The functional specification shall include rationale that the TSF is completely represented.

Evaluator action elements:

- ADV_FSP.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- ADV_FSP.3.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

6.2.2.7 ADV_HLD.3 Semiformal high-level design

Developer action elements:

- ADV_HLD.3.1D The developer shall provide the high-level design of the TSF.

Content and presentation of evidence elements:

- ADV_HLD.3.1C The presentation of the high-level design shall be semiformal.
- ADV_HLD.3.2C The high-level design shall be internally consistent.
- ADV_HLD.3.3C The high-level design shall describe the structure of the TSF in terms of subsystems.
- ADV_HLD.3.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.
- ADV_HLD.3.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

- ADV_HLD.3.6C The high-level design shall identify all interfaces to the subsystems of the TSF.
- ADV_HLD.3.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
- ADV_HLD.3.8C 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.
- ADV_HLD.3.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

Evaluator action elements:

- ADV_HLD.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- ADV_HLD.3.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

6.2.2.8 ADV_IMP.2 Implementation of the TSF

Developer action elements:

- ADV_IMP.2.1D The developer shall provide the implementation representation for the entire TSF.

Content and presentation of evidence elements:

- ADV_IMP.2.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.
- ADV_IMP.2.2C The implementation representation shall be internally consistent.
- ADV_IMP.2.3C The implementation representation shall describe the relationships between all portions of the implementation.

Evaluator action elements:

- ADV_IMP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- ADV_IMP.2.2E The evaluator shall determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements.

6.2.2.9 ADV_LLD.1 Descriptive low-level design

Developer action elements:

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

Content and presentation of evidence elements:

- ADV_LLD.1.1C The presentation of the low-level design shall be informal.
- ADV_LLD.1.2C The low-level design shall be internally consistent.
- ADV_LLD.1.3C The low-level design shall describe the TSF in terms of modules.
- ADV_LLD.1.4C The low-level design shall describe the purpose of each module.
- ADV_LLD.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.
- ADV_LLD.1.6C The low-level design shall describe how each TSP-enforcing function is provided.
- ADV_LLD.1.7C The low-level design shall identify all interfaces to the modules of the TSF.
- ADV_LLD.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.
- ADV_LLD.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.
- ADV_LLD.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

- ADV_LLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- ADV_LLD.1.2E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

6.2.2.10 ADV_RCR.2 Semiformal correspondence demonstration

Developer action elements:

- 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:

- 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.
- 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:

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

6.2.2.11 ADV_SPM.3 Formal TOE security policy model

Developer action elements:

- ADV_SPM.3.1D The developer shall provide a TSP model.
- ADV_SPM.3.2D The developer shall demonstrate or prove, as appropriate, correspondence between the functional specification and the TSP model.

Content and presentation of evidence elements:

- ADV_SPM.3.1C The TSP model shall be formal.
- ADV_SPM.3.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.
- ADV_SPM.3.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.
- ADV_SPM.3.4C 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.
- ADV_SPM.3.5C Where the functional specification is semiformal, the demonstration of correspondence between the TSP model and the functional specification shall be semiformal.
- ADV_SPM.3.6C Where the functional specification is formal, the proof of correspondence between the TSP model and the functional specification shall be formal.

Evaluator action elements:

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

6.2.2.12 AGD_ADM.1 Administrator guidance

Developer action elements:

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

Content and presentation of evidence elements:

- AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
- AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.
- AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.
- AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.
- AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.
- AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
- AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.
- AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

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

  Application Note: this assurance requirement helps in meeting the **SO.AdequateDocumentation** security objective, for the part pertaining to the administrator documentation.

6.2.2.13 AGD_USR.1 User guidance

Developer action elements:

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

Content and presentation of evidence elements:

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

Evaluator action elements:

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

  Application Note: correct use of the MIX by the user is of capital importance for the enforcement of the TSP. The user guidance should be written accordingly. This assurance requirement helps in satisfying the **SO.AdequateDocumentation** system objective.

6.2.2.14 ALC_DVS.1 Identification of security measures

Developer action elements:

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

Content and presentation of evidence elements:

- ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
- ALC_DVS.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

Evaluator action elements:

- ALC_DVS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

6.2.2.15 ALC_LCD.2 Standardised life-cycle model

Developer action elements:

- ALC_LCD.2.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.
- ALC_LCD.2.2D The developer shall provide life-cycle definition documentation.
- ALC_LCD.2.3D The developer shall use a standardised life-cycle model to develop and maintain the TOE.

Content and presentation of evidence elements:

- ALC_LCD.2.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.
- ALC_LCD.2.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.
- ALC_LCD.2.3C The life-cycle definition documentation shall explain why the model was chosen.
- ALC_LCD.2.4C The life-cycle definition documentation shall explain how the model is used to develop and maintain the TOE.
- ALC_LCD.2.5C The life-cycle definition documentation shall demonstrate compliance with the standardised life-cycle model.

Evaluator action elements:

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

6.2.2.16 ALC_TAT.2 Compliance with implementation standards

Developer action elements:

- ALC_TAT.2.1D The developer shall identify the development tools being used for the TOE.
- ALC_TAT.2.2D The developer shall document the selected implementation-dependent options of the development tools.
- ALC_TAT.2.3D The developer shall describe the implementation standards to be applied.

Content and presentation of evidence elements:

- ALC_TAT.2.1C All development tools used for implementation shall be well-defined.
- ALC_TAT.2.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.
- ALC_TAT.2.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

Evaluator action elements:

- ALC_TAT.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- ALC_TAT.2.2E The evaluator shall confirm that the implementation standards have been applied.

6.2.2.17 ATE_COV.2 Analysis of coverage

Developer action elements:

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

Content and presentation of evidence elements:

- ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.
- ATE_COV.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

Evaluator action elements:

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

6.2.2.18 ATE_DPT.2 Testing: low-level design

Developer action elements:

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

Content and presentation of evidence elements:

- 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:

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

6.2.2.19 ATE_FUN.1 Functional testing

Developer action elements:

- ATE_FUN.1.1D The developer shall test the TSF and document the results.
- ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation of evidence elements:

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

Evaluator action elements:

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

6.2.2.20 ATE_IND.3 Independent testing - complete

Developer action elements:

- ATE_IND.3.1D The developer shall provide the TOE for testing.

Content and presentation of evidence elements:

- ATE_IND.3.1C The TOE shall be suitable for testing.
- ATE_IND.3.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements:

- ATE_IND.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- ATE_IND.3.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.
- ATE_IND.3.3E The evaluator shall execute all tests in the test documentation to verify the developer test results.

6.2.2.21 AVA_CCA.1 Covert channel analysis

Developer action elements:

- AVA_CCA.1.1D The developer shall conduct a search for covert channels for each information flow control policy.
- AVA_CCA.1.2D The developer shall provide covert channel analysis documentation.

Content and presentation of evidence elements:

- AVA_CCA.1.1C The analysis documentation shall identify covert channels and estimate their capacity.
- AVA_CCA.1.2C 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.
- AVA_CCA.1.3C The analysis documentation shall describe all assumptions made during the covert channel analysis.
- AVA_CCA.1.4C The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.
- AVA_CCA.1.5C The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.

Evaluator action elements:

- AVA_CCA.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- AVA_CCA.1.2E The evaluator shall confirm that the results of the covert channel analysis show that the TOE meets its functional requirements.
- AVA_CCA.1.3E The evaluator shall selectively validate the covert channel analysis through testing.

6.2.2.22 AVA_MSU.2 Validation of analysis

Developer action elements:

- AVA_MSU.2.1D The developer shall provide guidance documentation.
- AVA_MSU.2.2D The developer shall document an analysis of the guidance documentation.

Content and presentation of evidence elements:

- AVA_MSU.2.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
- AVA_MSU.2.2C The guidance documentation shall be complete, clear, consistent and reasonable.
- AVA_MSU.2.3C The guidance documentation shall list all assumptions about the intended environment.
- AVA_MSU.2.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).
- AVA_MSU.2.5C The analysis documentation shall demonstrate that the guidance documentation is complete.

Evaluator action elements:

- AVA_MSU.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- AVA_MSU.2.2E 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.
- AVA_MSU.2.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.
- AVA_MSU.2.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

6.2.2.23 AVA_SOF.1 Strength of TOE security function evaluation

Developer action elements:

- AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

Content and presentation of evidence elements:

- AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.
- AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

  Application Note: this assurance component has two main identified purposes:

  - identify a Strength of Function regarding the cryptographic operation components of the TOE parts (this requirement is also included in [Iac99b]), and
  - identify a Strength of Function regarding the minimum *size* of the cryptographic keys used by the TOE.

  As a further note to the first point, it is useful to point out that as available computational power rises with time, also the strength of the cryptographic and probabilistic algorithms in the TOE must rise, to avoid direct attack and decryption of the transiting messages. This document will not state a particular strength of function (such as key length): these parameters must be chosen accordingly to available technology and perceived threats.

6.2.2.24 AVA_VLA.3 Moderately resistant

Developer action elements:

- AVA_VLA.3.1D The developer shall perform and document an analysis of the TOE deliverables searching for ways in which a user can violate the TSP.
- AVA_VLA.3.2D The developer shall document the disposition of identified vulnerabilities.

Content and presentation of evidence elements:

- AVA_VLA.3.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.
- AVA_VLA.3.2C The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.
- AVA_VLA.3.3C The evidence shall show that the search for vulnerabilities is systematic.

Evaluator action elements:

- AVA_VLA.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
- AVA_VLA.3.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.
- AVA_VLA.3.3E The evaluator shall perform an independent vulnerability analysis.
- AVA_VLA.3.4E 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.
- AVA_VLA.3.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a moderate attack potential.

Glossary

CA Certification Authority

CC Common Criteria for Information Technology Security Evaluation.

CCE Covert Channel Elimination

DoS Denial of Service.

EAL Evaluation Assurance Level

f.c. functional component

MMPP Multiple MIX Protection Profile

MUDAC Mandatory MIX User Data Access Control

PP Protection Profile

TOE Target Of Evaluation

TSC TSF Scope of Control

TSF TOE Security Functions

TSFI TSF Interface

TSP TOE Security Policy

SE Software Engineering

SF Security Function

SFP Security Function Policies

SMPP Single MIX Protection Profile

ST Security Target

References

[Cha81] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[Cot] L. Cottrell. Mixmaster & remailer attacks. http://www.obscura.com/~loki/remailer/remailer-essay.html.

[CYC97] D.W. Chadwick, A.J. Young, and N.K. Cicovic. Merging and extending the PGP and PEM trust models – the ICE-TEL trust model. *IEEE Network*, pages 16–24, May 1997.

[FGJP98] E. Franz, A. Graubner, A. Jerichow, and A. Pfitzmann. Modelling mix-mediated anonymous communication and preventing pool-mode attacks. In G. Papp and R. Posch, editors, *Global IT Security, Proceedings of the XV IFIP World Computer Congress*, pages 554–560. Austrian Computer Society, September 1998.

[FRE] The freedom network architecture (version 1.0). http://www.zks.net/products/whitepapers.asp.

[Iac99a] G. Iachello. Protection profile for an unobservable message delivery application using mixes. Revision 1.7, June 1999.

[Iac99b] G. Iachello. Single mix protection profile. Revision 1.11, May 1999.

[Iac99c] Giovanni Iachello. Software Evaluation Criteria and Information Technology Security. Master's thesis, Dipartimento di Elettronica e Informatica, Università degli Studi di Padova – Abteilung Telematik, Institut für Informatik und Gesellschaft, Universität Freiburg, June 1999.

[JTC98a] ISO JTC1/SC27/WG3. *Common Criteria for Information Technology Security Evaluation*, October 1998. Version 2.0 / ISO FDIS 15408.

[JTC98b] ISO JTC1/SC27/WG3. *Common Criteria for Information Technology Security Evaluation*, May 1998. Version 2 / CCIB-98-026.

[JW98] W. Jansen and J. Walsh. U.S. Government Application-Level Firewall Protection Profile for Low-Risk Environments. Technical report, NIST - NSA, August 1998. Version 1.a.

[RR98] Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, November 1998.