# A CC Approach to Windmill Control Systems

Shekoufeh Khodaverdi & Vikas Vohra

Kongens Lyngby 2007
IMM-THESIS-2007-13

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

# Abstract

This M.SC. thesis project evolves around Distributed Monitoring and Control (DMC) systems. The popularity of such systems has increased rapidly over the past decades, since it has become more and more necessary to control large systems of devices in a convenient and secure way.

The concrete case of a DMC system, that the thesis works through, is based on windmills. The access to the Windmill DMC System is provided by a web-based interface and all communication between entities in the system (web clients, web server, and windmills) is done over the Internet.

This thesis outlines a secure design proposal for systems of this kind, by analysing and defining the security requirements for such systems. The approach taken in order to do so is based on the Common Criteria for Information Technology Security Evaluation (CC). A CC Protection Profile (PP) is developed for a general Windmill DMC System. This PP is then used as basis for development of a Security Target (ST) document for a more specific Windmill DMC System. Finally, a design proposal is given in form of an Implementation Representation that illustrates the applicability of the ST for a concrete Windmill DMC System.

**Keywords:** Common Criteria, Protection Profile, Security Target, Distributed Monitoring and Control System (DMC system), Implementation Representation, Security, Windmills.

# Resumé

Dette eksamensprojekt omhandler distribuerede kontol- og overvågningssytemer (DMC). Anvendelsen af sådanne systemer er steget i popularitet gennem de sidste årtier, da der er opstået et behov for at kunne overvåge og styre store systemer over afstand på en sikker og nem måde.

Projektet tager udgangspunkt i en konkret case, som omhandler styring og overvågning af vindmøller. Vindmølle DMC systemet er tilgængeligt ved et web-baseret interface og al kommunikation mellem komponenterne i systemet (web klienter, web server og vindmøller) foregår over internettet.

Der er i dette projekt givet et udkast til et design til et sikkert vindmølle DMC system ved at analysere og definere sikkerhedskrav til sådanne systemer. Metoden der er benyttet til dette er baseret på Common Criteria (CC) standarden for evaluering af sikkerheden af IT systemer[1]. Der er udarbejdet en CC Protection Profile (PP) for et generelt vindmølle DMC system. PP'en er efterfølgende benyttet som basis til at udvikle et Security Target (ST) dokument for et mere specifikt vindmølle DMC system. Endeligt, for at vise ST'ens anvendelighed, er der udarbejdet et design forslag (implementation representation), som er formålet med dette projekt.

**Nøgleord:** Common Criteria, Protection Profile, Security Target, Distribueret kontrol- og overvågningssystem, DMC systemer, vindmølle, Implementation Representation, sikkerhed.

[1] Common Criteria for Information Technology Security Evaluation

# Preface

The present thesis is written in connection with a M.SC. project prepared at the Department of Informatics and Mathematical Modelling (IMM), Technical University of Denmark (DTU) during autumn 2006 and winter 2007. The project has been supervised by professor Robin Sharp, IMM, DTU.

We would like to thank everyone who has made this M.Sc. project possible. Especially, we would like to thank Robin Sharp for devoting time and energy in giving his excellent supervision and guidance.

Kgs. Lyngby, 12th February 2007

Shekoufeh Khodaverdi
s011629

Vikas Vohra
s011665

# Contents

**Abstract** **i**

**Resumé** **iii**

**Preface** **v**

**Contents** **x**

**1 Introduction** **1**

1.1 Organisation . . . . . 2

1.2 Terminology . . . . . 3

**2 The Common Criteria** **5**

2.1 Short about CC . . . . . 6

2.2 Protection Profile (PP) . . . . . 9

2.3 Security Target (ST) . . . . . 10

2.4 CC Approach for Designing Secure Systems . . . . . 11

**3 The PP Target of Evaluation 15**

3.1 DMC Systems . . . . . 15

3.2 The general windmill DMC System Model . . . . . 16

3.3 TOE Data . . . . . 18

3.4 TOE System Devices and Roles . . . . . 19

**4 Protection Profile (PP) 23**

4.1 TOE Security Environment . . . . . 24

4.2 Security Objectives . . . . . 31

4.3 Security Requirements . . . . . 36

4.4 PP Conclusion and Comments . . . . . 54

**5 The ST Target of Evaluation 57**

5.1 The TOE model . . . . . 57

5.2 TOE Components . . . . . 58

5.3 The Data Flows in the TOE . . . . . 60

5.4 TOE Devices and Roles . . . . . 62

**6 Security Target (ST) 65**

6.1 TOE Security Environment . . . . . 66

6.2 Security Objectives . . . . . 68

6.3 Security Requirements . . . . . 71

6.4 TOE Summary Specification . . . . . 83

6.5 ST Conclusion and Comments . . . . . 85

**7 Implementation Representation/Design 87**

7.1 Design of the TOE . . . . . 87

7.2 Security Functionality . . . . . 90

7.3 Conformance Claim . . . . . 99

7.4 Design Conclusion and Comments . . . . . 99

**8 Conclusion and Comments 101**

**A Protection Profile (PP) 103**

A.1 PP Introduction . . . . . 103

A.2 TOE Description . . . . . 107

A.3 TOE Security Environment . . . . . 112

A.4 Security Objectives . . . . . 115

A.5 Security Requirements . . . . . 117

A.6 Rationale . . . . . 146

**B Security Target (ST) 163**

B.1 ST Introduction . . . . . 163

B.2 TOE Description . . . . . 167

B.3 TOE Security Environment . . . . . 173

B.4 Security Objectives . . . . . 177

B.5 Extended Components Definition . . . . . 178

B.6 Security Requirements . . . . . 178

B.7 TOE Summary Specification . . . . . 210

B.8 Rationale . . . . . 214

**Bibliography** **237**

Chapter 1

# Introduction

Over the past years the windmill industry has expanded quite rapidly, with windmills being installed in thousands every year across the world [10]. With this immense evolution in the windmill industry the demand for a system for monitoring and controlling windmills has arisen. It has been proposed that such systems should be developed in a distributed fashion, so windmills can be supervised fx. via a web-based interface, enabling the control to be done from anywhere in the world using the Internet as the media for communication. Such systems go under the notation *Distributed Monitoring and Control* (DMC) systems.
Using such a distributed feature, ie. open architecture, introduces several threats to the system. It is therefore essential that the design of the system is carefully planned from a security point of view.

This M.Sc. thesis project will outline a secure design for a Windmill DMC System. In order to identify the security requirements for the system, the standard *Common Criteria for Information Technology* will be used. In the development process the steps listed beneath will be worked through:

1. Analysing the security requirements of windmill DMC systems in order to develop a general Protection Profile (PP) for such systems.

2. Developing a Security Target (ST) for a specific windmill DMC system based on the PP.
3. Preparing an implementation representation containing concrete specifications and thereby a detailed design document that can be implemented in the context of a concrete application.

## 1.1 Organisation

This paper documents the progress of work done in association with the development of a secure design of a Windmill DMC System. The paper is built up in individual chapters and appendices as stated below:

**Chapter 1**
This chapter contains an introduction to the thesis project including motivation, problem description, and organisation of the thesis paper. Additionally, the terminology used throughout this paper will be described in form of a list of abbreviations.

**Chapter 2**
This chapter gives a short introduction to the Common Criteria standard (CC) including the approach used in the project for designing a secure Windmill DMC System.

**Chapter 3**
This chapter contains a description of the Target of Evaluation (TOE). In order to get a description of a general Windmill DMC System, DMC systems in general are outlined. The TOE description in this chapter has to be seen in relation to the PP development.

**Chapter 4**
The PP is presented and explained in this chapter.

**Chapter 5**
The TOE is specified further for ST development in this chapter. The TOE is a more detailed version of the TOE described in chapter 3.

**Chapter 6**
The ST is presented and explained in this chapter.

**Chapter 7**
In this chapter the Implementation Representation is described and explained. A secure design proposal for a Windmill DMC system is presented in the context of a concrete application.

**Chapter 8**
Conclusions and final comments are stated in this chapter.

**Appendix A**
This appendix contains the PP document.

**Appendix B**
This appendix contains the ST document.

## 1.2 Terminology

### 1.2.1 Abbreviations

| | |
|---|---|
| **AES** | Advanced Encryption Standard |
| **CBC** | Cipher Block Chaining |
| **CC** | Common Criteria for Information Technology |
| **CORBA** | Common Object Request Broker Architecture |
| **CRC** | Cyclic Redundancy Check |
| **DAC** | Discretionary Access Control |
| **DBMS** | Database Management System |
| **DES** | Data Encryption Standard |
| **DMC** | Distributed Monitoring and Control |
| **EAL** | Evaluation Assurance Level |
| **FIPS** | Federal Information Processing Standard |
| **FTP** | File Transfer Protocol |
| **GIOP** | General Inter-ORB Protocol |
| **HTTP** | Hyper Text Transfer Protocol |
| **HTTPS** | HTTP using SSL |
| **ICE** | Internet Communications Engine |
| **IT** | Information Technology |
| **MAC** | Mandatory Access Control |

| | |
|---|---|
| **MAC** | Media Access Control address |
| **ORB** | Object Request Broker |
| **OS** | Operating System |
| **OSP** | Organisational Security Policy |
| **PKI** | Public Key Infrastructure |
| **PP** | Protection Profile |
| **RBAC** | Role-Based Access Control |
| **RC4** | Rivest Cipher 4 |
| **RSA** | Ron **R**ivest, Adi **S**hamir and Len **A**dleman Algorithm |
| **SAR** | Security Assurance Requirement |
| **SFP** | Security Function Policy |
| **SFR** | Security Functional Requirement |
| **SHA** | Secure Hash Algorithm |
| **SHS** | Secure Hash Standard |
| **SMTP** | Simple Mail Transfer Protocol |
| **SSL** | Secure Socket Layer Protocol |
| **ST** | Security Target |
| **TLS** | Transport Layer Security Protocol |
| **TOE** | Target of Evaluation |
| **TSF** | TOE Security Functionality |
| **TSFI** | TSF Interface |
| **TSP** | TOE Security Policy |
| **WDMC** | Windmill DMC System |
| **WPP** | WDMC System CC Protection Profile |

Chapter 2

# The Common Criteria

For owners and users of an IT system it is often wished to secure data in the system from being read, modified or deleted by unauthorised entities. Additionally the system is strongly wished to work appropriately and without any malfunction. Hence it is aimed to preserve the CIA principal: *confidentiality*, *integrity*, and *availability*. In order to be able to judge wether this principal is met by an IT system, consumers, developers, or evaluators of the IT system order an analysis of the security. This is also called a security evaluation.

Today the Common Criteria (CC) standard is widely used for evaluation of Secure Information Technology systems.

In this chapter there will be a short description of the Common Criteria standard and an explanation of how the CC approach can be applied in context with software engineering in order to design a secure system.

## 2.1 Short about CC

The CC is an international standard (ISO/IEC 15408) for computer security. The CC origins from the unification of the ITSEC[1], CTCPEC[2], and TCSEC[3] standards, so companies selling computer products for defence or intelligence use would only need to have them evaluated against one set of standards. The CC was developed by the governments of Canada, France, Germany, the Netherlands, the UK, and the US [11].

The CC is comprised of 3 parts [2]:

**Part 1:** Introduction and general model

**Part 2:** Security functional components

**Part 3:** Security assurance components

A more detailed description of these parts can be found in [2], [4], and [6].
The CC is primary used in context with development, evaluation and/or procurement of products with IT security functionalities, where the CC serves as a useful guide.
The way the CC is applied is by producing a Protection Profile (PP) and a Security Target (ST) document based on a product which within the CC context is called the Target of Evaluation (TOE). The contents of these documents are described in more details later in this chapter.

When buyers of a product want to be sure that the security level of the product meets their needs and therefore want to confirm or disprove the claims of the developers, following procedure is taken when applying the CC:

1. An organisation, that wants to procure a particular type of secure IT product, develop a PP in which their needs and requirements for security are stated. If the PP is evaluated as complete, consistent, and technically sound then it is published ([2] section 9.2).

2. The developer of the product takes the PP and develops a ST that they claim is in compliance with the requirements stated in the PP. The ST is

[1] ITSEC is the European standard, developed in the early 1990s by France, Germany, the Netherlands, the UK and also used by some other countries, e.g. Australia [11].
[2] CTCPEC is the Canadian Trusted Computer Product Evaluation Criteria [11].
[3] TCSEC is the United States Department of Defense standard, called the Orange Book and part of the Rainbow Series [11].

then evaluated (also called a ST evaluation[4]).

**3.** If the ST is approved the developer can then develop the product and have it evaluated against the ST (a TOE evaluation[5]).

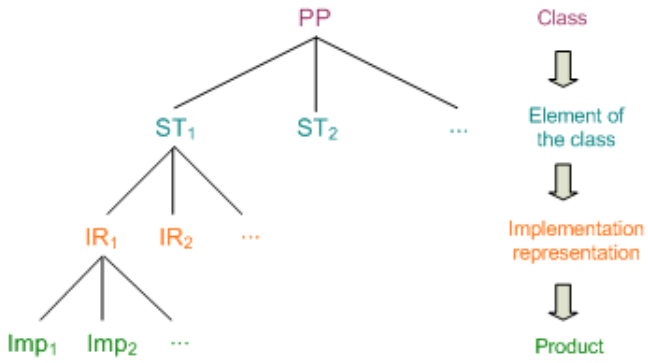


Figure 2.1: Summary of the CC approach.

| | **Consumers** | **Developers** | **Evaluators** |
|---|---|---|---|
| **Part 1** | Use for background information and reference purposes. Guidance structure for PPs. | Use for background information and reference purposes. Development of security specifications for TOEs. | Use for background information and reference purposes. Guidance structure for PPs and STs. |
| **Part 2** | Use for guidance and reference when formulating statements of requirements for a TOE. | Use for reference when interpreting statements of functional requirements and formulating functional specifications for TOEs. | Use as mandatory statement of evaluation criteria when determining whether a TOE meets claimed security functional requirements (SFRs). |
| **Part 3** | Use for guidance when determining required levels of assurance. | Use for reference when interpreting statements of assurance requirements and determining assurance approaches of TOEs. | Use as mandatory statement of evaluation criteria when determining the assurance of TOEs and when evaluating PPs and STs. |

Table 2.1: Road map to Common Criteria

Figure 2.1 describes this procedure. Additionally it can be seen that from one PP it is possible to develop several different STs, and from STs several different

[4]A Security Target evaluation is determination of whether the ST is complete, consistent, and technically sound and hence suitable for use as the basis for the corresponding TOE evaluation ([2] section 7.3).

[5]A TOE evaluation is determination of whether the TOE meets the TSP (TOE Security Policy) in the ST and whether the development environment of the TOE meets the SARs as specified in the ST ([2] section 7.3).

implementation representations. In the same way there can be several different implementations of the same implementation representation.
Table 2.1, which is taken from the official Common Criteria Part 1 [2], describes furthermore for which purposes the target audiences (consumers, developers, and evaluators) can use the 3 parts of the CC.

### 2.1.1 The CC framework

Unlike other standards, which list a set of requirements that should be acted upon when developing a secure computer system, the CC provides a framework that enables consumers to specify their security requirements, developers to show which security requirements their product adheres, and the evaluators to examine the developers' claims.

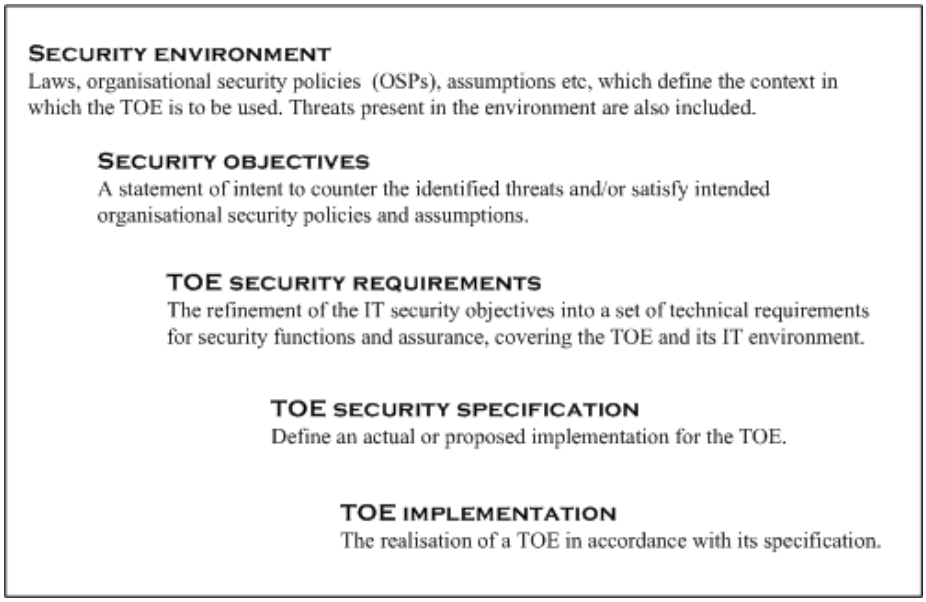


Figure 2.2: The framework of security concepts and terminology in CC is hierarchical.

The framework of security concepts and terminology, which the CC uses to discuss security, is built up hierarchical (see figure 2.2) [1]. As can be seen the CC framework is built up of 5 steps composed of analysis, measures, security requirements, and finally a concrete design (ie. an implementation representation) and if desired an implementation. This framework is applicable for both

the Protection Profile (PP) and the Security Target (ST).

In order to apply the CC the developers of the PP and ST statements have to define which entities of value in the TOE have to be secured. These entities are called assets.
Assets in the development environment are entities that the developer of the product places value upon whereas assets in the operational environment are entities that the owner of the TOE places value upon ([2] p. 12).

## 2.2 Protection Profile (PP)

As defined in Part 1 of the CC a Protection Profile is:

> *"An **implementation-independent** statement of security needs for a Product type."*

In other words a PP allows interested parties to express their security needs, and furthermore to provide a basis for writing a ST.
A PP is a general description of a specific type of TOE (ie. not a specific product but a product group), e.g. firewalls or smart card platforms. Hence PPs are reusable.
A PP is a template for a ST document that describes a specific product (see section 2.3).
Groups interested in writing PPs could among others be a user community that wants to work out a common set of security requirements for a given TOE type, developers of the same type of TOE that wish to define a minimum baseline for which security needs the TOE should meet, or governments or large corporations who want to procure IT products with specific requirements.
A PP must contain following sections:

1. **Introduction:** Narrative description of the TOE.
2. **Conformance Claim:** Conformance claims to other PPs.
3. **Security Problem Definition/Security Environment:** Description of threats, OSPs, and assumptions.
4. **Security Objectives:** The solution to security problems divided between the TOE and the operational environment of the TOE.
5. **Extended Components Definition:** Description of new components.

6. **Security Requirements:** Translation of security objectives for the TOE and for the development environment into the form of CC Part 2 and 3 requirements, ie. Security Functional Requirements (SFRs) and Security Assurance Requirements.

The content of these sections are described in details in [2] sections B.4 - B.9.

## 2.3 Security Target (ST)

The definition of a Security Target in CC Part 1 is as follows:

> *"An **implementation-dependent** statement of security needs for a specific identified TOE."*

Hence a Security Target statement is made by developers of products and describes the security specification of a specific product (e.g. Zone Alarm Firewall Pro 2005).
In the ST the sufficiency[6] of the TOE is analysed by following the steps described in figure 2.2. Thus a ST statement has to contain following sections:

1. **Introduction:** Narrative descriptions of the TOE on 3 levels of abstraction.
2. **Conformance Claim:** Conformance claims to PPs and/or packages.
3. **Security Problem Definition/Security Environment:** Description of threats, OSPs, and assumptions.
4. **Security Objectives:** The solution to security problems divided between the TOE and the operational environment of the TOE.
5. **Extended Components Definition:** Description of new components.
6. **Security Requirements:** Translation of security objectives for the TOE and for the development environment into the form of SFRs[7] and SARs[8].

[6] A TOE is sufficient when it does its assigned part (in conjunction with the other countermeasures in the operational environment) in countering the threats to assets in the operational environment ([2] section 7.1.1).
[7] These should be in form of CC Part 2 requirements and extended functional requirements.
[8] These should be in form of CC Part 3 requirements and extended assurance requirements.

7. **TOE Summary Specification:** Description of implementation of the SFRs.

More detailed description of the content of these sections are to be found in [3] section A.2.

## 2.4 CC Approach for Designing Secure Systems

In the previous sections the CC as a standard for evaluation of IT products was described. In this section it will be outlined how the CC will be used as a methodology for design of a secure system.
The approach adopts the idea from traditional software engineering which is an iterative process of development that starts with an abstract specification, moves on to a concrete specification and ends up with an actual design. Within the context of the CC the abstract specification is equivalent to the PP, the concrete specification to the ST, and the detailed design to the actual TOE implementation representation.
The iterative principle comes into play for instance

1. when moving from one development phase to another, e.g. from abstract specification to concrete specification (or vice versa), and/or

2. within a development phase it self, e.g. when analysing threats, assumptions, and security objectives, in the PP/ST.

The first case is quite obvious, since it can occur that you get new ideas and views when working on a later phase and thus it might be necessary to iterate back to a previous phase for revision.
Secondly, iterations can also occur within a phase (PP, ST or final design) itself. This is intuitively clear since a development phase consists of many sub-phases (see figure 2.3).
Figure 2.3 illustrates the steps involved in designing the system from a CC point of view[9].
The first step in the design process is to develop a PP based on a thorough analysis of security requirements for a Windmill DMC System, ie. the sub-phases as shown on the left in figure 2.3.
Next step is to derive a ST from the PP so threats, assumptions, OSPs are augmented and SFRs are refined for a specific system.

[9]The drawing is a reproduction of figure 1 in [15].

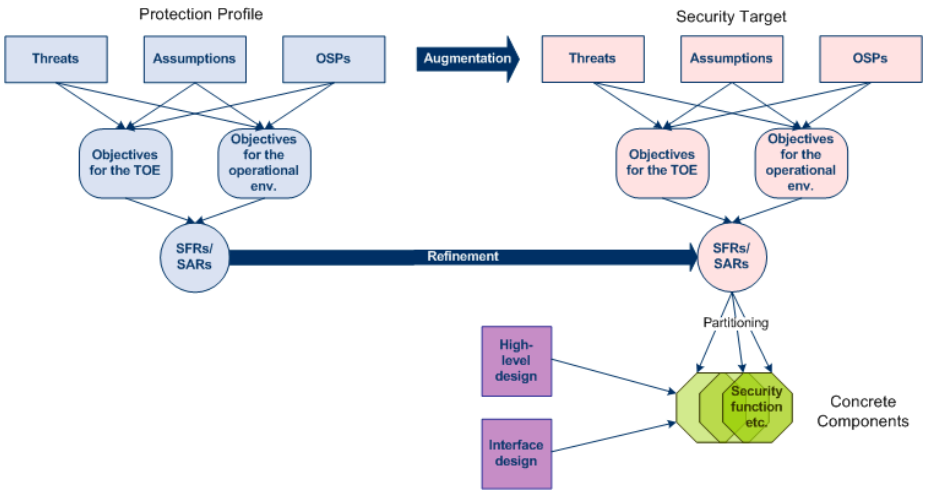


Figure 2.3: The development process of a secure system using the CC.

With the SFRs refined for a concrete Windmill DMC System in the ST, it is possible to produce a detailed design, ie. an implementation representation which can be used to build an actual application upon.
It is important to emphasise that the CC is going to be used as a design methodology and not its intended evaluation use. This reflects especially when using part 2 and 3 of the CC, since they contain SFRs and SARs which purpose is to base an evaluation upon.
The design and evaluation approaches are sketched in figure 2.4. The red arrows illustrate the relation between objectives and SFRs/SARs. In the original use SFRs and SARs are used for checking that objectives are satisfied. While in the design approach the SFRs and SARs are stated to implement and fulfil objectives and thereby state requirements to TOE.
Moreover, when identifying the evaluation assurance level (EAL) of a TOE ie. the scale for measuring the assurance that the TOE meets its security components and criteria for evaluation of PPs and STs, the EAL level is chosen on the basis of resources in terms of money and time.
Related to the evaluation process, the EAL level of a product is determined by customers. For instance a company might have certain expectations of security for products before even considering whether to purchase the product or not. Of course the EAL level also reflects the commercial value a buyer puts on products. Higher EAL implies higher purchase price.
The EAL level during design is something that is decided by the developer of the product and reflects how much guaranteed assurance the developer feels is in the product implementing its security requirements.

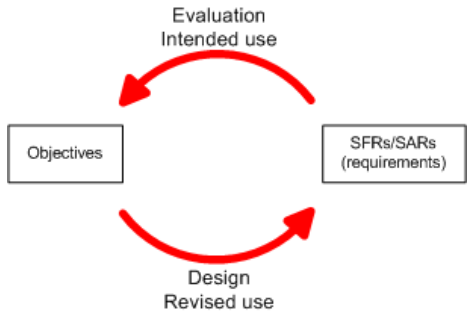


Figure 2.4: The CC SFRs and SARs used during design.

With that remark the approach used for designing a secure system using the CC as design methodology is sketched and can commence.

Chapter 3

# The PP Target of Evaluation

This chapter serves the purpose of introducing the windmill *Distributed Monitoring and Control* (DMC) system. Firstly a brief description of DMC systems will be given, followed by a more general DMC system model and its concrete use in the windmill scenario. This description will be the description of the Target of Evaluation (TOE) that the Protection Profile (PP) document will be based upon.
A more specific description of the TOE will be given when developing the Security Target (ST).

## 3.1 DMC Systems

The whole idea behind DMC systems is to monitor and control devices distributed geographically. DMC systems offer the advantage of centralised control, which is why their popularity has increased rapidly within recent time. Monitoring and controlling a lot of devices from a control centre is highly preferable rather than having to maintain several devices at a time. This also provides the opportunity to make devices operate synchronised in order to obtain better results. For instance windmills can be optimised in order to produce the desired amount of electricity by adjusting electricity production on different mills

through regulation of various windmill attributes from one place.

Furthermore DMC systems also give an excellent opportunity for systematically keeping track of any changes in data. This is due to the fact that all data concerning requests about the status or control of devices is stored in a *data trail*. The data trail thereby provides an overview of the activity and the responsibles for the activities in the system. The amount of data that should be stored depends on the actual use of the system. This is also applicable for length of time and quality of data to be stored.
The data trail is the backbone of such DMC systems since it supplies definite evidence of activities in the system. If it is lost the system will definitely lose its value, become more vulnerable against threats, and in worst case break down.

So the purpose of any DMC system is to:

1. Monitor and control devices in the system in a distributed way;
2. Keep record of changes in the system, ie. assuring proof/evidence through the data trail.

Often the transfer of data and commands in such DMC systems use some form of Ethernet or closed architecture in which a high level of security can be maintained. If instead an open architecture, like the Internet, is deployed for transmission it would make the system more vulnerable to attacks and threats. But on the other hand it would make it more versatile by also allowing access to the system outside the control centre, e.g. an inspector of devices out in the field [26] and any place with Internet connection.

## 3.2 The general windmill DMC System Model

With a brief idea of how DMC systems work and what they do, a general model of the DMC system in this project will be described. The general model will be used as basis for developing a PP (see section 2.2).
The DMC model in this project is built up such that the TOE can be accessed both from a central control centre and out in the field. This means the general DMC model will be accessed through an interface provided by a web service over the insecure Internet media.

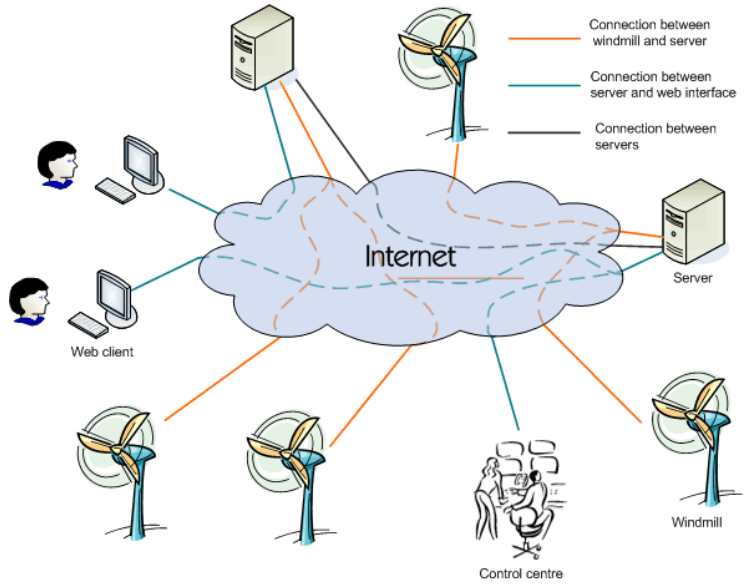


Figure 3.1: The overall structure of the TOE system consisting of web clients, servers, and windmills.

In figure 3.1 the distributed feature of the windmill control system can be seen. As shown on the figure users can access data about windmills via servers. The servers provide monitoring and control services through a web-interface. Servers can also be interconnected in order to get data about windmills they do not have information about. The Internet is used as media to connect entities in the system.
The way operations are performed rely on which specific action (ie. monitor or control) is performed and by which user role (see section 3.4). The operations the DMC system should satisfy are:

1. Monitoring and control devices in the system in a distributed way and
2. keeping record of changes in the system, ie. assuring proof/evidence through the data trail.

The flow of data in the system when operations are performed is sketched in figure 3.2. As shown, the data flows can be categorised into following:

- DMC input/output device - server

- Server - data trail
- Server - windmill

Users can through DMC devices generate a data flow to a server containing requests about monitoring or control of a windmill. The server processes the request by sending a request to the windmill in question. The windmill replies with a response which is again processed by the server and passed on to the output device of the user. Evt. the servers interact in order to get in connection with the windmill in question (this is also described previously in this section). Upon receiving requests and responses the server interacts with the data trail by reading/writing into the data trail.
Related to the operations of the DMC system, both requests and responses can be regarded as monitoring or control actions by users. While as the record into the data trail operation is the interaction between a server and the data trail.

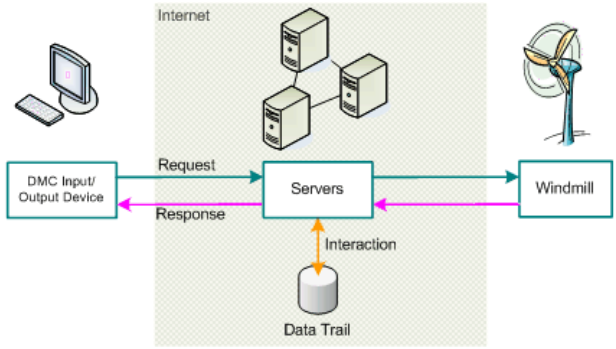


Figure 3.2: Data flows in the general DMC model.

## 3.3 TOE Data

In general data can be grouped into user data and data that is related to the security functionality of the TOE. User data is made by and used for users of the TOE and includes monitoring and control data since these operations are performed by users. Whereas data related to the security functionality (such as authentication data and audit records) of the TOE is made and used by the TSF (TOE security functionality) in order to assure TOE security. The TSF is that part of the TOE which is in charge of all security within the TOE. Its goal

is to implement all security functional requirements (SFRs).
Figure 3.3 reproduced from [5] p. 21 shows this division of TOE data very well. With this division of TOE data, the data that the data trail contains can be categorised into being partly user data (windmill monitoring and control data) and partly TSF data (audit records).

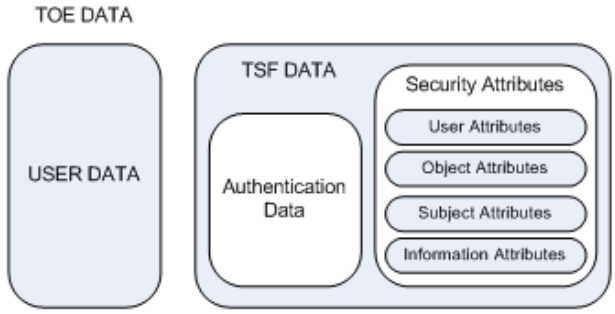


Figure 3.3: Relationship between user data and TSF data.

## 3.4 TOE System Devices and Roles

The actions that can be taken in the general DMC model can be performed by different entities using different devices. In this section a brief overview of the roles and devices that can access the DMC system will be given.

### 3.4.1 TOE Devices

The flow of data in the general DMC system is caused by actions being executed on input/ouput devices. The following list of devices is an outline of potential DMC input/output devices:

- Computers and laptops
- PDAs
- Smartphones
- etc.

It is important to take into account which devices are used in the DMC system, since some devices may be at greater risk than others and thus jeopardise the security of the system differently.

### 3.4.2 TOE Roles

Another factor to consider is the roles users of the DMC system are assigned. Each role interacts in its own individual way with the system and thereby uphold individual rights and permissions when security is concerned.
Following roles are identified:

**Service technician -** The role of a service technician is to make sure that apparatus and instruments of the devices work correctly, ie. responsible in maintaining the physical security and has nothing to do with the IT functionality. In order to judge whether there is any malfunction in the system, it has to be possible for the technician to read (monitor) the status of the devices in the DMC system. The workplace of a technician in the Windmill DMC System is at the windmills out in the field.

**Operations engineer -** The operations engineer's job is to monitor and control the DMC system. Depending on regional divisions, number of operations engineers, workload, etc. rights and privileges to access data of the DMC system can vary from engineer to engineer. In other words operations engineers may monitor and control a subset of the system. Operations engineers are furthermore responsible for validating that both monitoring data and control data in the DMC system are correct. The operations engineer can perform the tasks from either the control centre or out in the field.

**Administrator -** The administrator is the one who is responsible for the overall functionality and security of the DMC system. The administrator of the DMC system owns rights and privileges to perform changes (installation and configuration) in order to maintain the functions and security of the DMC system. In addition the administrator is responsible of user accounts, ie. creation of new user profiles with appropriate rights and privileges as well as maintenance of already existing user profiles.

The roles of the DMC system could be either static or dynamic, ie. the rights and privileges of a role can change over time or not. A dynamic model would be more suitable if frequent occurrence of promotion/degradation and thereby change in range of responsibilities among roles is present.

When possessing a role it is obviously clear that individuals that own that role are competent and trustworthy to carry out the work and responsibility that is demanded.

Chapter 4

# Protection Profile (PP)

This chapter contains the considerations made during development of the Protection Profile (PP). As stated in chapter 2, the steps in preparing a PP can be seen in figure 4.1.

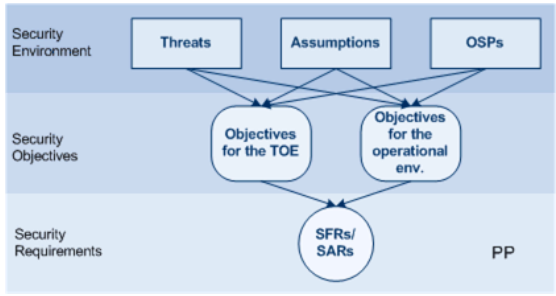


Figure 4.1: The PP process.

The contents of this chapter should be read in correspondence to the PP which can be found in appendix A. It is important to emphasise that this chapter contains the discussions and argumentations made when developing the PP, while as the document in appendix A is the actual technical PP paper.

## 4.1 TOE Security Environment

In this section the upper layer of the PP, ie. the security environment (see figure 4.1) in which the TOE is to operate, will be derived. The way it will be done is by analysing the threat scenario. The threats, assumptions, and OSPs will be outlined in this section. The section will end with a brief discussion on amount of security contra costs and other available resources which play a significant role when obtaining security in a system.

### 4.1.1 Threats to Security

In order to develop a proper PP in accordance with the CC, a threat scenario must be set up. This includes identifying assets to be protected, possible threat agents that could harm the TOE and the specific threats that could be performed.

#### 4.1.1.1 Assets

Identification of assets that need to be protected is necessary in order to specify threats and vulnerabilities of the TOE. The assets are derived directly from the purpose of the TOE. Following operations were identified for a general DMC system:

1. Monitor and control devices in the system in a distributed way.
2. Keep record of changes in the system, ie. ensure proof/evidence through the data trail.

From this the assets that need protection are the data trail, monitoring data, and control data. Any malicious modification in these data jeopardises the security of the TOE. In the PP monitoring and control data will be considered as user data and data trail considered as TSF data (see section 3.3).
Below is listed an overall description of the possible critical points in the model, where the assets are most vulnerable:

**Servers -** The data trail, that resides on servers, is valuable to the TOE since it contains significant information in order to uphold the security of the TOE. Furthermore all data received and sent through the servers are also

vulnerable. Servers (and thereby all the assets) are vulnerable to system breakdown, physical damage or attacks by malicious users or programs.

**Windmills -** Windmills are other potential weakness points in the TOE. It is expected that windmills send correct monitoring data when requested and upon receiving control data they act correspondingly. Within a windmill monitoring data may be read incorrectly due to malfunction of monitoring apparatus or data may be modified by unauthorised entities. Furthermore, received correct control data may be modified by malicious activity in the windmill.

**Input/output devices -** These devices are one of the critical points in the model because they might be infected with malicious programs that could harm the TOE when users gain access to the TOE through these devices.

**Connections -** Since the connections in the TOE are established over an insecure media, they are potential targets for attacks. Both monitoring data and control data are threatened by data loss, being read and/or modified.

#### 4.1.1.2 Threat Agents

Additional to specification of assets, it is necessary to identify entities that would have interest in comprising the security of the TOE. In principal any entity could pose a threat to the system. These entities could do harm either unconsciously or because they have evil intensions. The entities include persons as well as programs.
The threat agents are divided into 2 groups: *internal attackers* and *external attackers*.

**Internal attackers** are entities within the company itself.

**External attackers** are correspondingly entities outside the company borders.

Hereinafter the term *attacker* will cover both groups of threat agents.
There could be several reasons for wanting to break into the TOE and gain access to valuable TOE data. Among these could be jealousy, competition, industrial espionage, revenge, or fun.
This leads to the observation that a threat agent can be characterised by the factors such as expertise, available resources, and motivation [14]. It is obvious that an entity that is involved of all three factors is of greater threat to the TOE than an entity with lack of one or more of the factors. Observations show that the strongest factor is motivation. An entity with high motivation and a

given level of expertise and a set of resources is more likely to launch an attack compared to another entity that has lower motivation but the same expertise and resources [14].
Having said that, the factors expertise and resources do not have so much impact on whether an entity launches an attack or not, ie.:

> *low expertise + resources + high motivation ≈ high expertise + resources + high motivation*

and

> *less resources + expertise + high motivation ≈ more resources + expertise + high motivation.*

#### 4.1.1.3 Threats

Threats must be identified and stated in the PP. In the following an overview of potential threats are listed and explained:

**T.MASQUERADE** ◇ *Unauthorised user or process pretends to be another entity in order to gain access to data or other TOE resources.*

The TOE is threatened by users or any malicious processes that steal or gain access to data and/or information contained in TOE by impersonating an authorised entity. Gaining another entity's access information could be done by sniffing, eavesdropping, backdoors, etc. Man-in-the-middle attacks are also included in this threat.

**T.UNAUTHORISED_ACCESS** ◇ *Mischievous users or programs may gain unauthorised access to data which they are not allowed to according to the TOE security policy.*

Gaining privileges and rights to access data that the user is not authorised for, as stated clearly in the OSP (see section 4.1.3), poses a threat to the TOE system. The unauthorised user would in worst case be able to view, modify or delete critical data in the TOE.

**T.MODIFICATION** ◇ *Attackers may try to maliciously fiddle with protected data of the TOE.*

This threat deals with the problem of data manipulation. An attacker may do modification, deletion, or any other evil intended action to TOE data. If this becomes actual, it would seriously jeopardise the security of the system and leave it unreliable and question its integrity.

**T.UNATTENDED_SESSION** ◇ *An attacker may gain unauthorised access to an unattended session.*

Unattended sessions can be exploited by attackers to gain unauthorised access and perform malicious activity to the TOE. Such threats often occur when authorised users become unaware and/or their attention is distracted to something else, and they leave their workstation. This threat is specially relevant since monitoring and control of windmills is done by a web interface that is available from anywhere with a connection to the Internet.

**T.ACCIDENTAL_USER_ERROR** ◇ *Users may make accidental errors that could jeopardise the security of the TOE.*

Users that are to operate the TOE most likely possess different capabilities and competencies. This threat takes into account that users might use the system in a wrong way and make accidental errors.

**T.DATA_TRANSMISSION** ◇ *An attacker may alter the transmission and thereby the confidentiality and the integrity of the data in the TOE.*

No matter if it is input or output data that is altered during transmission, it will compromise the security of the system. If data is modified or even lost during transmission it could lead to severe consequences. In the windmill case, this would at worst lead to breakdown of the components. Since the transmitted data contains sensitive data (windmill data, passwords, etc.) the threat must also cover confidentiality of data.

**T.CRYPTO_LEAK** ◇ *Key data or other executable code associated with the cryptographic functionality, which intends to protect the data in the TOE system, may be viewed, modified or deleted by mischievous users or programs.*

By encrypting data in the TOE it is aimed at achieving confidentiality, integrity, and accessability. If data about the cryptographic functionality is accessed by mischievous users or programs, it may be exploited to steal or modify data in the TOE. For instance if cryptographic keys are stolen it could be possible for attackers to read confidential data and/or use the keys to launch attacks.

With a thorough description of the threat scenario for the TOE in hand, it can be considered what precautions and thereby which objectives the TOE should satisfy in order to be secure in relation to the threat scenario (see section 4.2).

### 4.1.2 Secure Usage Assumptions

Assumptions are related to terms that can not be directly referred through IT items and thus the TOE can not implement or enforce these. Assumptions are stated in a PP since they contribute in giving a full description of the environment in which the TOE will be deployed.
The following descriptions identify the assumptions needed for the TOE to be securely operational.

**A.CORRECT_DEVELOPMENT** ◇ *The development of the TOE, ie. design, implementation, and test, is assumed to be carried out correctly so it results in a TOE without flaws and errors that may lead to exploration by malicious users or programs.*

If the TOE is designed or implemented poorly or even tested insufficiently, errors in the TOE can be spotted and used for malicious activities in the TOE by hackers or other mischievous users or programs.

**A.NO_EVIL** ◇ *It is assumed that administrators have no evil intentions and that they are appropriately trained to carry out their job correctly.*

Administrators are responsible of the overall functionality and security of the TOE and have extended privileges in the system so they have to be well-intended, competent, and follow administrator guidance to achieve a secure TOE.

**A.PHYSICAL** ◇ *The physical security of TOE is assumed provided in order to avoid physical loss or damage of the TOE due to external factors like fire, theft, natural catastrophes etc. Thus by this assumption the security of the data and the functionality of TOE are preserved.*

In order to uphold the physical security of the TOE it is assumed that all the physical parts of the TOE are provided a secure physical environment.

The assumptions listed above could actually be stated as threats (which also was done originally in this project). But since there can not be taken full precautions in order to meet the threats they are instead stated as assumptions.

For instance the assumption A.CORRECT_DEVELOPMENT was originally divided into three threats that respectively covered poor design, poor implementation, and poor test. Although it is possible to minimize errors and flaws during development of an IT product by using specific standard procedures, it is impossible to find a solution for avoiding these errors and flaws entirely.
Likewise the assumption A.PHYSICAL must be made since it is possible for instance that a user drops a laptop or that a tornado causes damages to windmills. It is not possible to safe guard the TOE against these scenarios.

### 4.1.3 Organisational Security Policies (OSPs)

The OSPs are a set of rules, practices and procedures imposed by the organisation to address security needs. Following OSPs are identified:

**P.AUTHORISED_USERS** ◇ *The TOE can only be accessed by authorised users.*

By restricting access to the TOE by authentication and identification this policy ensures that only allowed users get access to the functionality of the TOE. This could be realised by password protection, biometrics or some other authentication and identification mechanism.

**P.USER_PRIVILEGES** ◇ *Users have different rights and privileges to access TOE data.*

This policy ensures that users of the TOE are granted rights and privileges according to their respective needs and functionality. For instance in a company a manager would have permission to access certain documents that an ordinary employee may/would not. This policy is associated with the user roles identified in section 3.4.2 since it states clearly which data in the TOE are accessible by which roles.

**P.ACCOUNTABILITY** ◇ *Users that are authorised access to TOE data shall be held accountable for their actions within the TOE.*

This policy states that all activity in the TOE should be accounted for. This is to ensure that records are stored so that if something unpredicted or any deliberate malicious actions happen, the responsible user will be held accountable.

**P.CRYPTOGRAPHY** ◇ *Data in the TOE has to be encrypted following some standard cryptographic algorithms.*

Encryption is necessary in order to ensure secure data. Thus this policy must be present. The policy includes encryption/decryption services as well as key management, signatures, etc.

**P.TRAIN** ◇ *Authorised users of the TOE shall be trained appropriately in operating the TOE.*

In order to ensure that users get the necessary skills to work securely with the functions within the TOE, some training shall be given in advance.

Almost every organisation that uses IT has some policies that intend to ensure security. With the above listed policies it is possible for an organisation that wish to purchase the TOE to judge whether the TOE can be applied within their policy statements.
For instance the OSP P.USER_PRIVILEGES relies on the structure and size (thereby number of employees) of the organistation. Same reasoning is applicable for the other stated OSPs.
The contents and existence of OSPs depend to great degree on how much the organisation prioritises security of IT products.

### 4.1.4 Security vs. Resources

As with other aspects of security policy, what kind of security an organisation decides to implement should be a function of risk analysis and threat assessments. How much money, time and effort one is willing to spend on security depends on potential losses arising from its breach. For most companies this means, for instance in regards to physical security, things like locked server rooms, additional authentication or access controls, and possibly some kind of monitoring system to track access and use of sensitive systems. These can vary from simple logging mechanisms to video surveillance systems, depending on risk assessments and needs for accountability.
This means that the level and amount of security is very much dependent on available resources that the management of a company has to decide upon. Thus developers of a system have to take the amount of available resources in consideration when they design a secure system. In the following costs and resources will be taken into account as much as possible and it is emphasised not to draw or make up any unrealistic or exaggerated security measures with that in mind that no fixed amount is given in this project.

## 4.2 Security Objectives

Following will contain an overview of the security objectives that aim at countering the threats and/or comply with any OSPs and assumptions that were identified in section 4.1. This corresponds to the second layer of the development of a PP (see figure 4.1). Which threats, OSPs, and assumptions that are encountered for by each stated objective, is shown in table 4.1.

| | O.UNIQUE_IA | O.DATA_INTEGRITY | O.ACCOUNTABILITY_AND_AUDIT | O.CRYPTO_FUNCTIONS | O.ROLE_MANAGEMENT | O.SESSION | O.BACK-UP | O.VULNERABILITY_ANALYSIS | O.CRYPTO_SECRECY | O.SELF_TEST | OE.TRAIN | OE.ISOLATION |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T.MASQUERADE | × | × | × | | | | | | | (×) | | |
| T.UNAUTHORISED_ACCESS | × | × | × | | × | | | | | (×) | | |
| T.MODIFICATION | × | × | × | × | | | × | | | (×) | | |
| T.UNATTENDED_SESSION | | | | | | × | | | | (×) | × | (×) |
| T.ACCIDENTAL_USER_ERROR | | | × | | | | × | | | (×) | × | |
| T.DATA_TRAMSMISSION | | × | | × | | | | | | (×) | | |
| T.CRYPTO_LEAK | | | | | | | | | × | (×) | | |
| A.CORRECT_DEVELOPMENT | | | | | | | | × | | (×) | | |
| A.NO_EVIL | | | | | | | | | | (×) | × | |
| A.PHYSICAL | | | | | | | × | | | (×) | | × |
| P.AUTHORISED_USERS | × | × | × | | × | | | | | (×) | | |
| P.USER_PRIVILEGES | | | | | × | | | | | (×) | | |
| P.ACCOUNTABILITY | × | | × | | × | | | | | (×) | | |
| P.CRYPTOGRAPHY | | | | × | | | | | × | (×) | | |
| P.TRAIN | | | | | | | | | | (×) | × | |

Table 4.1: Mapping from security objectives to threats, assumptions, and policies.

**O.UNIQUE_IA** ◇ *The TSF shall ensure that unauthorised access to data in the TOE is not allowed. This shall be done by unique identification and authentication of entities trying to gain access to the TOE.*

This security objective mainly ensures identification and authentication of entities. This implies that unauthorised entities are not allowed access to the TOE and that users can not fake other users' identity.
Hereby T.MASQUARADE, T.UNAUTHORISED_ACCESS, and P.AUTHORISED_USERS are ensured for through this objective. The threat T.MODIFICATION is also covered because malicious and unauthorised modification of data is prevented, and also because if a malicious modification of data has occurred this objective can contribute to identify which entity has been causing this. Additionally this objective is abide by the P.ACCOUNTABILITY OSP since in order to obtain accountability it is first of all necessary to identify entities and thereby record their activities in the TOE.

**O.DATA_INTEGRITY** ◇ *Unauthorised modification, theft, or deletion of TOE data (user data and TSF data) shall be prevented.*

In order to maintain integrity in TOE data, this objective shall be ensured in the TOE by the TSF. The TOE data has to be adequately protected from unauthorised attempts to modify data.
From this it can be derived that T.MASQUERADE, T.UNAUTHORISED_ACCESS, T.MODIFICATION, T.DATA_TRANSMISSION, and P.AUTHORISED_USERS are countered for by this objective.

**O.ACCOUNTABILITY_AND_AUDIT_RECORDS** ◇ *The TSF shall provide individual accountability for audited events. The audit records shall record date and time of action and the identity of the entity responsible for the action.*

Individual accountability and audit records give knowledge about who has been granted access (including unauthorised access) to the TOE and which events each entity has caused. The records shall contain a description of the actions, the date and time the actions were carried out, and the identity of the user that caused the actions so that concrete evidence is provided. This objective hereby makes it possible to register breach in the security and if possible to face the responsible entities with the violations of the policies or rules.
By making use of audit records the policy P.ACCOUNTABILITY which states that any activity in the TOE has to be traced back to an entity, will be satisfied.
Furthermore if any unauthorised access has been detected in the audit records the administrators should be notified in order to fix the problem

and cover up the security hole and contribute in satisfying P.AUTHORISED_USERS. In addition the audit records shall only be read by users with special read access, ie. no modification or deletion of data is allowed due to the purpose audit records serve, ie. ensuring evidence.

This objective thereby counters the threats T.MASQUERADE, T.UNAUTHORISED_ACCESS, T.MODIFICATION, and T.ACCIDENTAL_USER_ERROR. Additionally the OSPs P.AUTHORISED_USERS and P.ACCOUNTABILITY are covered.

**O.CRYPTO_FUNCTIONS** ◇ *The TSF shall implement approved cryptographic algorithms.*

This security objective aims at protecting data of the TOE. It states that proper cryptographic measures shall be provided in securing the data. This includes securing data during transmission. Any cryptographic function in the TOE, ie. encryption/decryption, authentication, signature generation/verification, and key generation is dealt with by this objective. Therefore the objective counters T.MODIFICATION and T.DATA_TRANSMISSION, and ensures P.CRYPTOGRAPHY.

**O.ROLE_MANAGEMENT** ◇ *The TSF shall provide a mechanism for administrators to control rights and privileges according to user roles.*

In a system where roles change often (see section 3.4.2), ie. new entities are allowed access, existing roles are updated, or new rights and privileges are granted it is important to ensure a procedure for adjusting and maintaining rights and privileges of roles periodically. This includes account generation and account update.
This objective ensures T.UNAUTHORISED_ACCESS, P.AUTHORISED_USERS, P.USER_PRIVILEGES, and P.ACCOUNTABILITY.

**O.SESSION** ◇ *The TSF shall provide mechanisms that lock sessions automatically when the activity in an open session has been idle in a predefined period of time. Furthermore it shall be possible for users to manually lock a session in order to avoid signing out. Users shall be able to unlock a session by re-authentication and just continue the session where it was left.*

If a session has not been active over a period of time the TSF shall lock the session in order to avoid that unauthorised persons can take over the session. Additionally users that want to leave a session but not sign out in order to continue their work when they are back at their workstation, shall be able to do this. Re-authentication, when unlocking a session, is necessary so that the system can verify the user of the session.

This objective is relevant because of the threat posed by T.UNATTENDED_SESSION.

**O.BACK-UP** ◇ *The TSF shall provide procedures for back-up of TOE data. The data trail must be recoverable at any time.*

Back-up is an essential feature of the TOE so that if any physical loss should occur, data of the TOE is not lost, especially not the data trail. The purpose of this security objective is thus to ensure that tracing back and restoring data is possible and thus counters T.MODIFICATION and T.ACCIDENTAL_USER_ERROR. This security objective also covers physical damages/loss, ie. A.PHYSICAL is ensured.
The distributed architecture provides good facilities for back-up.

**O.VULNERABILITY_ANALYSIS** ◇ *The TSF will undergo vulnerability analysis in order to verify that design, implementation and test of the TOE does not contain any flaws.*

As discussed previously, it is not possible to completely ensure that design, implementation and test of a system is entirely flawless. Nevertheless in order to meet the assumption A.CORRECT_DEVELOPMENT this security objective has been identified. Regular vulnerability analysis will, if not entirely then close to so, identify flaws during design, implementation and test phase. At least obvious flaws of the TOE will be discovered.

**O.CRYPTO_SECRECY** ◇ *Key data or other executable code associated with the cryptographic functionality shall be kept secret.*

This objective aims at keeping cryptographic data (keys, signatures, algorithms, etc.) secret and thereby it prevents loss, theft or modification of these data.
The threat T.CRYPTO_LEAK and the OSP P.CRYPTOGRAPHY is covered by this objective.

**O.SELF_TEST** ◇ *The TOE shall provide self-testing functionality for all TOE security functions which can detect security vulnerabilities in the form of flaws and intrusions.*

During operational usage phase the TOE shall be able to run tests and scans for detection of any potential threats and vulnerabilities that may compromise the TOE security. In other words this objective aims to ensure a reactive control system that can detect and locate flaws and intrusions (e.g. worms, vira, and spyware), and as a reaction it shall initiate corrective actions in order to avoid or reduce damages. It is realised that a proactive procedure that can detect attacks or errors before they occur/show, is almost impossible. An example is a virus detection program

that is able to scan any kind of data that enters the TOE before it is saved on the hard disk. The antivirus program can not guarantee that no vira or worms will enter the system since new vira and worms may not have any definitions in the antivirus program and hence will not be recognised as malicious data.
The self test objective can also be seen as a procedure that follows a check list where security functions are evaluated. This assists in identifying vulnerabilities that can be safe-guarded against. It can also be a helpful procedure to discover security breaches when the TOE has undergone some changes (updates, add-ons, etc.).
Self testing could for instance include procedures for scan of files, looking through log-files, evaluation of physical state of the TOE, checking for and installing updates (e.g. antivirus programs, cryptographic algorithms and keys, and firewalls) and evaluation of competency of the personnel. So what the self test objective covers, is actually a discussion on whether it is able to satisfy any threats, assumptions, or policies fully. In order to keep things simple, it is decided that self test is included in this section but not in the technical PP found in appendix A. Additionally this objective shall be viewed as a supplement to the other objectives listed in table 4.1 in meeting threats, assumptions, and OSPs. Furthermore it should be noticed that this objective actually does not always encounter identified threats but ensures reduced damages, spread of damages, and guard against similar future attacks or flaws. Thus in table 4.1 the threats, assumptions, or OSPs that O.SELF_TEST can not guarantee but can have a preventive impact on, are marked with a cross in parentheses. These markings will not appear in the PP.

### 4.2.1 Environmental Security Objectives

Some security objectives are not directly related to IT but are aimed at the environment in which the TOE is to operate. The environment has great influence on the security of the TOE since threats, identified in section 4.1.1, can be countered by observing and taking necessary precautions in the environment itself. These environmental security objectives are largely satisfied through procedural or administrative measures.

**OE.TRAIN** ◇ *Training of administrators, operational engineers, and service technicians will be provided by the overall responsible of the TOE.*

Training is a necessary part of any organisation and system in order for the staff to be competent, trustworthy, and kept updated. The training

includes guidance to secure usage of the TOE and awareness of existing and newly added security policies (e.g. policies about not leaving an open session or having passwords floating around).
This objective addresses T.ACCIDENTAL_USER_ERROR, T.UNATTENDED_SESSION, P.TRAIN, and A.NO_EVIL.

**OE.ISOLATION** ◇ *Those responsible for the TOE shall provide isolation of physical parts of the TOE such that they are protected from physical damages, intrusion, and theft.*

The physical protection can be met by isolation of for instance servers, control rooms, and windmills (apparatus). This can be done by for example using safety rooms or sensitive systems.
The assumption A.PHYSICAL is taken care of by this objective. Besides this the threat T.UNATTENDED_SESSION is also partly addressed due to the fact that an isolated control room will not allow unauthorised entities to take over any up and running sessions. But this objective will not be sufficient for cases where sessions are run on for instance laptops which can not be isolated out in the field.

With the security objectives of the TOE in hand, it is possible to move on to the next stage in the PP where security requirements for the TOE have to be specified.

## 4.3 Security Requirements

The security requirements needed in order to fulfill security objectives will be identified and discussed in this section. The section is built up in three subsections, each addressing respectively: security functional requirements, requirements for the TOE environment, and security assurance requirements.

### 4.3.1 Security Functional Requirements (SFRs)

This section identifies the appropriate security functional requirements (SFRs) the TOE shall meet in order to satisfy the security objectives stated in section 4.2 (except for the OEs). This together with identification of assurance requirements in section 4.3.3, correspond to the final lower layer in developing the PP (see figure 4.1).
In order to identify which SFRs are needed to satisfy objectives part 2 of CC

[5] will be used. A summary of included SFRs is listed in table 4.2.
The considerations and reasonings behind selection of SFRs shown in the table will be described and discussed below. As can be seen, some of the SFRs are specified for several objectives. This is because some SFRs can cover different aspects of security and thus concern several different objectives at the same time.
Notice that no environmental objectives are included in the table since they can not be satisfied by SFRs but instead need to be taken care of by non-IT functional administrative procedures. These are discussed in section 4.3.2.

#### 4.3.1.1 Unique Identification and Authentication

The security objective O.UNIQUE_IA will be satisfied by following SFRs.

***FIA_UID.2 - User identification before any action***
This SFR is found suitable since identification is required before any actions are allowed. The CC states that this component is hierarchical to FIA_UID.1, which allows specified actions to be performed before the user is identified. But this should not be the case according to the objective description, which ensures identification of all users and accountability of their actions, and thus FIA_UID.2 is chosen.

***FIA_UAU.2 - User authentication before any action***
For the same reasons as for FIA_UID.2, this component is chosen in preference to FIA_UAU.1, ensuring authentication and accountability of all user actions.

***FIA_UAU.3 - Unforgeable authentication***
The main reason for including this SFR is to ensure that **unique** authentication and thereby also identification is present in the TSF. The SFR provides a mechanism that is able to detect and prevent use of authentication data that has been forged or copied. Thus this SFR ensures the aspect of unique identification and authentication of the O.UNIQUE_IA objective.

***FIA_UAU.6 - Re-authenticating***
Since one of the identified objectives of TOE is the O.SESSION where re-authentication is necessary in order to unlock a locked session this SFR is needed. In this SFR the conditions under which re-authentication has to be carried out, are specified. Same rules for ordinary identification and authentication apply for re-authentication, ie. above mentioned directives.

| | O.UNIQUE_IA | O.DATA_INTEGRITY | O.ACCOUNTABILITY_AND_AUDIT | O.CRYPTO_FUNCTIONS | O.ROLE_MANAGEMENT | O.SESSION | O.BACK-UP | O.VULNERABILITY_ANALYSIS | O.SELF_TEST | O.CRYPTO_SECRECY |
|---|---|---|---|---|---|---|---|---|---|---|
| FAU_ARP.1 | | | × | | | | | | | |
| FAU_GEN.1 | | | × | | | | | | | |
| FAU_GEN.2 | | | × | | | | | | | |
| FAU_SAA.1 | | | × | | | | | | | |
| FAU_SAR.1 | | | × | | | | | | | |
| FAU_SAR.2 | | | × | | | | | | | |
| FAU_STG.1 | | | × | | | | | | | |
| FCS_CKM.1 | | | | × | | | | | | × |
| FCS_CKM.2 | | | | × | | | | | | × |
| FCS_CKM.4 | | | | × | | | | | | × |
| FCS_COP.1 | | | | × | | | | | | × |
| FDP_IFC.1 | | × | | | | | | | | |
| FDP_IFF.1 | | × | | | | | | | | |
| FDP_ITT.1 | | × | | | | | | | | |
| FDP_SDI.1 | | × | | | | | | | | |
| FIA_UAU.2 | × | | | | | × | | | | |
| FIA_UAU.3 | × | | | | | | | | | |
| FIA_UAU.6 | × | | | | | × | | | | |
| FIA_UID.2 | × | | | | | | | | | |
| FMT_MOF.1 | × | × | × | × | × | × | × | × | × | × |
| FMT_MSA.1 | × | × | × | × | × | × | × | × | × | × |
| FMT_MSA.2 | | | | × | | | | | | × |
| FMT_MSA.3 | | × | | | | | | | | |
| FMT_MTD.1 | × | × | × | × | × | × | × | × | × | × |
| FMT_SMF.1 | × | × | × | × | × | × | × | × | × | × |
| FMT_SMR.1 | × | × | × | × | × | × | × | × | × | × |
| FPT_AMT.1 | | × | | | | | | | | |
| FPT_ITT.1 | | × | | | | | | | | |
| FPT_RCV.2 | | | | | | | × | | | |
| FPT_STM.1 | | | × | | | | | | | |
| FPT_TST.1 | | × | | | | | | | × | |
| FTA_SSL.1 | | | | | | × | | | | |
| FTA_SSL.2 | | | | | | × | | | | |

Table 4.2: SFRs satisfying security objectives.

#### 4.3.1.2 Accountability and Audit Records

The objective O.ACCOUNTABILITY_AND_AUDIT is ensured best by the FAU class that covers security audit issues. From this class following families and/or components are identified to be covering the objective in the best possible way.

***FAU_ARP.1 - Security alarms***
When intrusion or violation of TOE security functions has been detected, the TSF has to take some actions in order to stop the intrusion or correct the security violation. Responsible persons should also be informed. This component is dependant on the inclusion of FAU_SAA.1.

***FAU_SSA.1 - Potential violation analysis***
In order to have a security alarm functionality in the TSF it is necessary that the TSF knows what to react upon in case of violation of security functions. With this SFR a set of rules in monitoring the audited events is specified, which then will be used by the TSF to indicate a potential violation.

***FAU_GEN.1 - Audit data generation***
This component of the security audit data generation (FAU_GEN) family defines requirements for the level of auditable events, and specifies the list of data that shall be recorded in each record [5] (ie. subject-object binding).

***FAU_GEN.2 - User identity association***
This is another component of the FAU_GEN family that ensures that the TSF is able to associate each auditable event with the identity of the user that caused the event [5].

***FAU_SAR.1 - Audit review***
Since audit records are evidence of what has been going on in the TOE and can be used to detect unwanted activity it shall be possible for administrators to read audit information from the audit records.

***FAU_SAR.2 - Restricted audit review***
In accordance with the O.ACCOUNTABILITY_AND_AUDIT objective responsible users shall have access to view audit records. No one is allowed to modify or delete audit records since the audit data is very significant for the maintenance of the TSF security (see section 4.2).

***FAU_STG.1 - Protected audit trail storage***
This component ensures protection of audit records from unauthorised access, modification, and/or deletion.

***FPT_STM.1 - Reliable time stamps***
In order to ensure reliable time stamps for auditing issues and security attribute expiration, this SFR is needed. Furthermore this component is included because the FAU_GEN.1 and FAU_GEN.2 components are dependant on the existence of reliable time stamps.

#### 4.3.1.3 Management

In order to uphold the security of the TOE, the different aspects of the TSF that need to be managed, have to be identified and necessary management specifications have to be stated. The nature of the TOE and the environment it is to operate in implies that it is a dynamic system in regards to management of security attributes, functions, and data. In contrast to a static system where these security values are determined once and can not be changed, a dynamic system makes it possible for administrators to redefine the values when necessary.
When deciding upon which parameters in the TSF that should be manageable careful considerations must be thought of. When permitting a lot of parameters to be manageable the system is easier to adapt to new states (new users, redefinition of user roles, changing cryptographic algorithms, etc.) but because of the large number of parameters it will be more difficult to control and manage. In contrast to this few manageable parameters means harder adaption and easier control. This leads to considerations on which of the factors adaptability or controllability is put weight upon in the system when identifying which and how many parameters should be managed.
For the TOE in mind, a dynamic system is most appropriate because it has to be able to adapt to new states where the structure of the organisation and the functionality of the system can vary over time. So adaptability is prioritised higher in this system. This means parameters having influence on all objectives will be manageable and therefore some of the selected SFRs cover all objectives, specially those SFRs related to security functions.
The management of the TSF is mainly dealt with in the FMT (Security management) class. This class identifies 3 aspects of the TSF: security attributes, TSF data, and TSF functions [5]. Additionally the FMT class can be used to specify management roles and their interaction.

***FMT_MOF.1 - Management of security functions behaviour***
This component specifies restrictions to the behaviour of authorised users (roles) when managing the behaviour of security functions in the TSF [5]. In other words this component specifies security functions and which users have the privileges to manage these.
From the specified roles in the TOE, it is obvious that only administrators shall

have privileges to manage the behaviour of security functions in the TSF. The functions that have to be managed by the administrators are:

*a) Functions implementing creation and recovery of back-ups;*

*b) Functions implementing role management, including administration and maintenance of delegated roles;*

*c) Functions implementing routines for identifying events that have to be audited, and administration and maintenance of audit records;*

*d) Functions implementing methods for identification and authentication of users;*

*e) Functions implementing and maintaining access control methods;*

*f) Functions implementing methods for locking sessions;*

*g) Functions implementing secure procedures for data transfer;*

*h) Functions implementing procedures for assuring physical security and its maintenance;*

*i) Functions implementing methods for self test and analysis of results from the tests;*

*j) Functions implementing timers and clock synchronisation; and*

*k) functions for managing any cryptography related issues.*

***FMT_MSA.1 - Management of security attributes***
The security attributes (such as access control, audit records, and cryptographic functions) of the TSF must be manageable. This SFR includes capabilities for viewing and modifying these attributes for specific roles identified in section 3.4.2. For obvious reasons the security attributes must be restricted to be available for administrators only.

***FMT_MSA.2 - Secure security attributes***
With this SFR it is ensured that only secure values are accepted for security attributes.

***FMT_MSA.3 - Static attribute initialisation***
The TSF shall enforce the information flow control SFP (Security Function Policy) to provide either restrictive or permissive default values for security attributes that are used to enforce the SFP.
Only administrators are allowed to specify alternative initial values to override

the default values when an object or information is created ([5] p. 108).

***FMT_MTD.1 - Management of TSF data***
This component restricts the ability to access the TSF data for authorised users. These users possess special rights which enables them to read, modify, or delete TSF data. Hereby it shall be specified that only administrators are allowed to manage:

*a) Data trail;*

*b) Identification and authentication data;*

*c) Cryptographic algorithms and keys; and*

*d) audit records.*

***FMT_SMF.1 - Specification of Management Functions***
The TSF functions that have to be managed are:

*a) Functions to create and recover back-ups;*

*b) Functions to administrate and maintain delegated roles*

*c) Functions to maintain audit records and to identify events that have to be audited and administrated;*

*d) Functions to identify and authenticate users;*

*e) Functions to protect data by using access control methods;*

*f) Functions setting up and maintaining session locking attributes;*

*g) Functions that provide secure procedures for data transfer;*

*h) Functions assuring physical security and its maintenance;*

*i) Functions for self testing and analysing results from the testing;*

*j) Functions to synchronise timers and clock; and*

*k) functions that manage cryptography related issues.*

#### 4.3.1.4 Role Management

The O.ROLE_MANAGEMENT objective is ensured by the security management (FMT) class, which specifies the management of different aspects of the TSF. In the following, components that seem to cover the different parts of this objective will be mentioned and shortly described.

***FMT_SMR.1 - Security roles***
The roles that are given to users have to be maintained by the TSF in order to keep the TSF secure. The roles as identified in section 3.4.2 are:

*a)* Service technician

*b)* Operations engineer

*c)* Administrator

Furthermore, the components that are described in section 4.3.1.3, also contribute in management of the roles since roles are described by security attributes.

#### 4.3.1.5 Session

Partly ensured by management requirements, the objective O.SESSION is also met by the SFRs listed below. The chosen components are needed in order to incorporate automatically and manually session locking. In order to manage security attributes of session locking some of the already described management SFRs (see section 4.3.1.3) are needed for the O.SESSION objective. These SFRs are the FMT_MOF.1, FMT_MSA.1, FMT_MTD.1, FMT_SMF.1, and FMT_SMR.1.

***FTA_SSL.1 - TSF-initiated session locking***
This component defines a session locking procedure carried out by the system. The period of time, that a session is allowed to be interactive before session locking, is also specified hereunder. The component specifies furthermore the events that should occur prior to unlocking of the session.

***FTA_SSL.2 - User-initiated locking***
This component defines a session locking procedure carried out by the user. The component specifies furthermore the events that should occur prior to unlocking

of the session.

Both FTA_SSL.1 and FTA_SSL.2 have dependency to FIA_UAU.1 (Timing of authentication) (see section 4.3.1.1 FIA_UAU.2). Furthermore the FIA_UAU.6 (Re-authenticating) SFR is necessary in order to cover O.SESSION.

#### 4.3.1.6 Cryptography

As such there are two security objectives identified that concern cryptography: O.CRYPTO_FUNCTIONS and O.CRYPTO_SECRECY. The first objective covers the functional use of cryptography and the second secrecy of cryptographic data such as keys.
When ensuring these objectives, by finding appropriate SFRs, it is not possible to look at the objectives separately since they are related, ie. some SFRs ensuring one objective depend on the presence of another SFR that satisfies another objective.
The CC provides the cryptographic support class (FCS) for assuring cryptography measures in the TOE. The components, taken from the families of the FCS class, satisfying the two objectives will be described.

***FCS_COP.1 - Cryptographic operation***
This SFR addresses the O.CRYPTO_FUNCTIONS objective since it concerns the cryptographic operations used in the TOE. Through this SFR any cryptographic operation is required to be performed in accordance with a specified algorithm and with a cryptographic key of specified size.
Of course by specifying the use of cryptographic keys FCS_COP.1 depends highly on how keys are managed in the TOE. Several dependencies (*FCS_CKM.1* and *FCS_CKM.4*) are therefore listed in the CC when including this SFR. Fortunately, these dependencies are provided for when ensuring the O.CRYPTO_ SECRECY objective.
Furthemore, this SFR is also dependent of the presence of secure security attributes (*FMT_MSA.2*) because clearly cryptographic data (such as keys) are security attributes in them selves, and thus only secure values are valid for these keys. The FMT_MSA.2 SFR is described in section 4.3.1.3.

In order to apply cryptography in any system cryptographic keys are necessary. It is important that these cryptographic keys are managed appropriately and securely through out their life cycle right from generation till destruction. Therefore appropriate requirements must be stated in order to manage keys securely.

***FCS_CKM.1 - Cryptographic key generation***
This SFR ensures that cryptographic keys are generated in accordance with a specified algorithm and key size. Generation of keys is the first step in managing keys and therefore must be done securely.
FCS_CKM.1 partly contributes to ensure objective O.CRYPTO_FUNCTIONS since it is not possible to ensure that cryptographic operations are performed properly if cryptographic keys are not appropriately generated.
Furthermore, FCS_CKM.1 also ensures O.CRYPTO_SECRECY since generation of keys must not be possible by other parties. Any forgery of keys has to be prevented and keys have to stay secret.

***FCS_CKM.2 - Cryptographic key distribution***
This component has been included since proper methods for distributing cryptographic keys among entities is a necessity in meeting O.CRYPTO_FUNCTIONS and O.CRYPTO_SECRECY. In order for the TOE components (ie. web clients, servers and windmills) to exchange data in encrypted format and to carry out mutual authentication, key sharing is needed among the components. Thus it is a requirement that cryptographic keys are distributed following an approved key distribution method.

***FCS_CKM.4 - Cryptographic key destruction***
This SFR takes into account the destruction of cryptographic keys. It ensures that keys are destroyed appropriately with a clear destruction method. Just like generation, destruction contributes in assuring both the objectives O.CRYPTO_FUNCTIONS and O.CRYPTO_SECRECY.
Destruction of cryptographic keys is a necessity in order to keep track of which keys are active and to avoid old keys getting into hands of unauthorised entities. If keys fall into wrong hands (say by theft) or an employee is dismissed and is seeking revenge, it is crucial that the keys are destroyed immediately since it is possible to perform all the cryptographic operations of the TOE with it still being active.
So it is important that destruction is done securely in order to satisfy O.CRYPTO_FUNCTIONS and O.CRYPTO_SECRECY.

#### 4.3.1.7 Data Protection

Since the CC considers data protection in 2 categories ie. user data and TSF data (see figure 3.3), likewise approach will be applied in following selection of SFRs which aim at achieving data protection of all TOE data and thereby satisfy the objective O.DATA_INTEGRITY.

**User Data Protection**

The CC class that provides the necessary requirements concerning protection of user data is FDP - User data protection. This class groups user data protection families into 4 groups [5]:

1) *User data protection security function policies:* The security function policies and their scope of control are specified in this group.

2) *Forms of user data protection:* This group deals with what kind of user data protection shall be provided, e.g. access control functions and information flow control functions.

3) *Off-line storage, import and export:* This group addresses the trustworthy transfer into or out of the TOE.

4) *Inter-TSF communication:* This group deals with the integrity and confidentiality of data when it is transferred within the TOE.

To specify which security function policies (SFPs) shall be used, it is necessary to look at what kind of access control technique is most suitable for the TOE. Access control techniques are intended to prevent accidental or malicious destruction of information, and furthermore controlling the release and propagation of that information. There are in general two categories of access control methodologies: discretionary and mandatory. Main access control techniques are described below.

**Discretionary Access Control (DAC)**

In DAC the access policy is entirely determined by owners of resources (e.g. files, directories, data, and system resources). Owners of resources decide to whom they want to give access. They can also control what other users can do with their resources (ie. read, write, delete, or execute). In other words DAC allows subjects to access objects solely based on the subjects' identity and the authorisation rule [16].
Since the access policy is in the hands of owners only, it is possible that accidental or malicious granting of access to data to entities, that should not be allowed access, can occur and this would compromise the security of the TOE.
Furthermore, it can be noticed that DAC only controls information release, not its propagation once released [27].

**Mandatory Access Control (MAC)**

In MAC the access policy is entirely determined by the system and thus

not by owners of resources as in DAC. MAC is usually used in multilevel[1] systems that process highly sensitive data, such as classified government and military information [11]. With MAC it is possible to implement a system where a subject is permitted to access an object only if the subject's security clearance dominates the security classification[2] of the object [16].

**Role-Based Access Control (RBAC)**
RBAC is a MAC which has been developed at NIST to meet the needs of industry [18]. When applying RBAC users of the TOE are assigned roles for various job functions. Access to data and permissions to perform certain operations are granted to specific roles and thus not to specific entities as in MAC. In other words permissions to perform certain operations are acquired through roles that have been assigned by an administrator. This technique results in easier management of individual user rights and privileges [11].

From these descriptions of access control techniques and the fact that roles are already defined in the TOE, it is decided that the RBAC access control technique will be most suitable for the TOE in mind. The reason for this choice is to be found in the fact that the access control policies for user data of the TOE have to be assigned by administrators of the TOE. In other words it is the TSF that has to control whether information may flow from a resource to a user. It has therefore been decided that the *information flow control policy (FDP_IFC)* is chosen instead of the *access control policy (FDP_ACC)*. The TSF mechanism in the information flow control policy is in [4] described as a mechanism that does not allow operations to change any security attributes since it would be in contradiction to an information flow control SFP.
The following components are considered most suitable as SFRs for user data protection in the TOE.

***FDP_IFC.1 - Subset information flow control***
This component specifies an information flow control SFP. Additionally, it defines the list of subjects (e.g. users, machines, or processes), information (e.g. email or network protocols), and a subset of the possible operations in the TOE that this policy shall be enforced upon.

***FDP_IFF.1 - Simple security attributes***
The TSF shall enforce the information flow control SFP specified in FDP_IFC.1 based on types of subject and information security attributes. The list of subjects and information controlled by the indicated SFP, and for each, the secu-

[1] A multilevel system is a single computer system that handles multiple classification levels between subjects and objects.
[2] The classification systems vary from country to country but most have levels corresponding to "Top Secret", "Secret", "Confidential", "Restricted", and "Unclassified" [11].

rity attributes are specified in this SFR. This component has a dependency to FMT_MSA.3 which is described in section 4.3.1.3.

***FDP_ITT.1 - Basic internal transfer protection***
This SFR ensures protection of user data when it is transferred within the TOE, ie. via internal channels. Since the TOE in mind is a distributed system and is composed of various physically-separated parts (see section 3.2) this component covers this kind of user data protection.

***FDP_SDI.1 - Stored data integrity monitoring***
In order to maintain the integrity of stored user data in the TOE this SFR is picked. It ensures protection of stored data by monitoring user data stored in containers controlled by the TSF for specified integrity errors on all objects, based on defined user data attributes. This SFR is relevant for the data trail which besides TSF data also contains stored user data.

**TSF Data Protection**

The class that addresses this area is FPT - Protection of the TSF. This class specifies SFRs that ensure integrity and management of TSF functionalities and integrity of TSF data.

***FPT_AMT.1 - Underlying abstract machine test***
The underlying abstract machine is a virtual or physical machine upon which the TSF executes. In order to verify the security assumptions made about the underlying abstract machine, such as memory capacity and correct mode of operation, this component specifies the conditions under which the verification has to occur by the TSF.
This SFR is related to the self test objective as well since testing the underlying abstract machine is part of the TSF self test.

***FPT_ITT.1 - Basic internal TSF data transfer protection***
This SFR ensures integrity of TSF data that is being transferred between physically-separated parts of the TOE via internal channels. This corresponds closely to FDP_ITT.1 that stated the same objective for user data.

#### 4.3.1.8 Self Test

In order to preserve security of the TOE the TSF has to periodically test its functionality and analyse whether it still is secure or not. This includes both detection of unauthorised entities (e.g. worms, vira, and spyware) and detection of flaws and errors in the different parts of the TSF (e.g. servers, file

systems, and sensors). The FPT class (protection of the TSF) seems to suit the O.SELF_TEST objective best since it focuses on protection of TSF data.

***FPT_TST.1 - TSF self test***
This component specifies conditions under which self test should occur and the integrity of which parts of the TSF should be verified. Thus the integrity and assurance of correct operation of TSF is preserved by this SFR. This component is dependant on FPT_AMT.1 described in section 4.3.1.7.

#### 4.3.1.9 Back-up

There are several ways of ensuring back-up in a system. SFRs found to fulfill the O.BACK-UP security objective will be presented and the reasoning behind including them will be discussed within the scope of the TOE.

***FMT_SMF.1 - Specification of Management Functions***
Among several other management functions, this SFR provides the means for an administrator to ensure continued operation of the TOE, including back-up and recovery. Thereby it specifies management functions for creating and recovering back-ups.
Back-up is a management function and therefore it is found adequate to include this management component in ensuring the O.BACK-UP security objective.

***FPT_RCV.2 - Automated recovery***
A thing to consider when implementing back-up and recovery is whether the mechanisms should be manual (e.g. carried out by an administrator) or automated. The choice of method depends on considerations on how much data loss affects the TOE, the amount of data to be backed up, and how regular back-ups should be carried out. Furthermore thoughts about how many resources (e.g. memory capacity and CPU power) back-up procedures demand, have to be made before deciding upon specific back-up solutions.
In the TOE considered in this project it is the data trail that has to be recoverable at any time. Back-up of data trail should be done on a regular basis since loss of data could mean costly damages in regards to windmills. Therefore an automated approach is to be preferred.
This SFR specifies a list of failures/service discontinuities that the TSF shall recognise and react upon automatically.
This SFR is dependant on the inclusion of the assurance requirement AGD_OPE.1 which also provides operational user guidance to back-up procedures. The assurance requirements can be found in section 4.3.3.

### 4.3.2 TOE Environment Requirements

The TOE operational environment contributes to the security of the TOE therefore it is important to consider how the security objectives for the environment can be met. This section concerns the issue of satisfying environmental objectives (OEs) that are defined in section 4.2.1. It is important to emphasise that requirements for the environment are not functional requirements which is the reason for why they are not included in table 4.2.
The CC states following about the security objectives for the operational environment:

> *There is no translation required in the CC for the security objectives for the operational environment, because the operational environment is not evaluated and does therefore not require a description aimed at its evaluation ([3] p. 59).*

So even though the CC states that translation for security objectives for the operational environment are not required in the PP, it is found relevant to address them anyway to ensure overall security of the TOE.

**OE.TRAIN**
In order even to be considered for operating the TOE, a user must have assigned a role, identified in section 3.4.2. This means that there must be some procedure for evaluation if a user can be assigned the role. These procedures should in principal be present in the assurance requirement called AGD_OPE.1 - Operational user guidance. It should specifically be stated how training shall be carried out in terms of which qualification tests users should go through, which certificates they should acquire, and which courses they should attend before they are assigned a role.

**OE.ISOLATION**
It was previously assumed (A.PHYSICAL) that physical protection was provided as a general assumption. Reasonable precautions like safety rooms, barriers and fences to sensitive parts of the TOE (see section 4.1.1) are expected to be in order.
Although physical protection of the TSF is addressed in the "TSF physical protection (FPT_PHP)" family, it does not state any measures for encountering the actual damage or theft of parts. The components in the family only detect when/if something physical to parts happen and specify what should be done. This is not sufficient to fulfil the objective. But together with the stated general assumption the OE is satisfied to satisfactory extent.

It is important to keep in mind that the specified requirements to the operational environment of TOE are guidance lines to reach a reasonable level of security in relation to cost and other resources available (see section 4.1.4). Furthermore, it must be noticed that it is not possible to measure or test whether or how much the requirements for the environment are followed.

### 4.3.3 Security Assurance Requirements (SARs)

The final step in completing the PP, is to identify the Security Assurance Requirements (SARs). The definition of assurance within CC scope is as follows:

> *Assurance - grounds for confidence that a TOE meets the SFRs ([3] p. 13).*

Together with the SFRs, the SARs thereby correspond to the lower layer shown in figure 4.1.
The CC part 3 defines a list of security assurance components that are the basis for the security assurance requirements. Each assurance component reflects an assurance requirement to be met.

The means by which assurance is obtained in CC context is by evaluation. But since assurance is to be used as part of a design process, the assurance is to be decided by developers of the IT product, as discussed in section 2.4. The way this is done is by choosing an appropriate level of assurance (EAL). The security assurance components and requirements are derived directly from the EAL. The CC provides a clear overview of EALs and corresponding assurance components that are needed at that particular EAL (see [6] section 8). The chosen EAL level is based on following reasoning:
Firstly, the two lowest assurance levels (EAL1 and EAL2) only reflect basic assurance. Secondly, to have a product at level 5 or higher (EAL5 - EAL7) it is needed to rely upon underlying systems, among other the operating system. This means that they also must have at least same assurance as the TOE itself. Since it is not within the scope of this project to analyse these underlying systems, these EALs are not considered for this TOE. This only leaves EAL3 or EAL4 to be considered.
When comparing the two levels, it is important to take into consideration what the purpose of the TOE is, and under which circumstances and environment it will be deployed (see section 3).
Furthermore, when looking closer at the assurance components that are different at the two levels, it is noticed that stronger demands during development, especially tests and vulnerability analysis which are identified as profound security

objectives and a vital part of the environment in which the TOE is deployed, are in greater focus at EAL4 (e.g. in the AVA_VAN family).
During development more assurance is given by EAL4 than EAL3 by requiring a design description, an implementation specification/representation, and improved mechanisms/procedures that provide confidence that the TOE will not be tampered with during development or delivery. Especially an outline for an implementation representation is exactly what is aimed for in this project.
Assessing the context in which the TOE is to operate further indicates the choice of EAL. The TOE is to operate within a rather closed environment by predefined known users/roles, ie. people with windmill knowledge and who are company authorised. Violation of security could have severe consequences financially and physically, and can affect the individual living being, because the windmills contribute electricity to the power system and their operation is important to the overall power supply[3]. So the TOE must ensure that windmills are functioning correctly. This means high assurance to its security is highly relevant.
Having said that, EAL4 would be the most likely choice, but there are some requirements defined in EAL4 which are beyond the scope of this project. This includes for instance considerations on how the TOE shall be delivered and demands for giving a subset of the actual implementation.
Therefore it is concluded that the level of assurance stated by EAL3 without any extra augmentation is found most appropriate and therefore chosen. Since a partly implementation representation of the TSF is aimed for in this project, the assurance requirement ADV_IMP.1 of EAL4 would have been ideal to include and thereby augment EAL3 with this component. But because ADV_IMP.1 has dependency on other components of EAL4, this is abstained from.
The included assurance components are predefined by the CC, and listed in table 4.3.
Notice that the assurance component AGD_OPE.1 covers all objectives. This is due to the given definition in [7] section 13.1 which states that this component is an operational user guidance document. It describes the security functionality provided by the TSF and gives instructions and guidelines, and helps to understand the TSF. Furthermore, it includes the security-critical information and actions required for its secure use.

[3]Much like the power circuit breakdown stated in [22].

| *Assurance Class* | *Assurance components/SARs* | |
|---|---|---|
| ADV:Development | ADV_ARC.1 | Security architecture description |
| | ADV_FSP.3 | Functional specification with complete summary |
| | ADV_TDS.2 | Architectural design |
| AGD:Guidance documents | AGD_OPE.1 | Operational user guidance |
| | AGD_PRE.1 | Preparative procedures |
| ALC:Life-cycle support | ALC_CMC.3 | Authorisation controls |
| | ALC_CMS.3 | Implementation representation CM coverage |
| | ALC_DEL.1 | Delivery procedures |
| | ALC_DVS.1 | Identification of security measures |
| | ALC_LCD.1 | Developer defined life-cycle model |
| ASE:Security Target evaluation | ASE_CCL.1 | Conformance claims |
| | ASE_ECD.1 | Extended components definition |
| | ASE_INT.1 | ST introduction |
| | ASE_OBJ.2 | Security objectives |
| | ASE_REQ.2 | Derived security requirements |
| | ASE_SPD.1 | Security problem definition |
| | ASE_TSS.1 | TOE summary specification |
| ATE:Tests | ATE_COV.2 | Analysis of coverage |
| | ATE_DPT.1 | Testing: basic design |
| | ATE_FUN.1 | Functional testing |
| | ATE_IND.2 | Independent testing - sample |
| AVA:Vulnerability assessment | AVA_VAN.2 | Vulnerability analysis |

Table 4.3: Security assurance components in EAL3[7].

## 4.4 PP Conclusion and Comments

A complete and general Protection Profile for windmill DMC systems has been developed according to the procedures outlined in chapter 2.4. The full PP can be found in appendix A.

The windmill DMC system, ie. the TOE, in this PP is based on a general model of DMC systems. Having said that, it should be kept in mind that the development of the PP has been specifically aimed at systems for monitoring and controlling windmills. And any other similar systems can claim conformance to this PP.

In order to develop the PP first of all a thorough threat analysis has been made in order to get a complete view of which threats, vulnerabilities, and threat agents can jeopardise the TOE security. The threat scenario together with analysis of usage assumptions and security policies provide the basis for defining security objectives for the TOE system. The objectives have been translated into a standardised language in form of requirements, ie. SFRs and SARs that should be enforced upon the TOE.

Development of the PP is an iterative process which means that in order to get a satisfactory result a lot of time needs to be devoted to add and review its contents. A typical iterative step was when identifying requirements (SFRs and SARs). While performing this task it was necessary to review the security objective that the requirement was to satisfy. All though a very time consuming task, it can be concluded that the more iterative steps that are made, the more a thorough PP is developed.

When identifying SFRs it was a general rule to select relevant and as few as possible components that could sufficiently cover the specified objectives. Having said that it should be mentioned that selecting SFRs is an open ended task which does not have any ultimate solution. In other words it is possible to select a lot of SFRs that seem nice to include as requirements but are really not essential since other SFRs could sufficiently cover the objectives. This also has impact on how extensive the PP is. If the PP is too extensive no ST will be able to claim conformance to it.

As assurance level EAL3 is found most suitable. This choice was made upon the context and environment in which the TOE is to operate, and the fact that some objectives were assured for directly by assurance components.

A general observation when designing secure systems is that security can not be ensured 100 %. The level of security is very much dependant on the amount

of resource you are willing to devote. Resources include time, costs, and human resources. The fact that security cannot be guaranteed 100 % is also reflected in the process of specifying objectives that shall encounter defined threats. Some threats are not possible to guard against but the damage they can cause can be limited and similar future attacks can be recognised and encountered.

Chapter 5

# The ST Target of Evaluation

This chapter will define the Target of Evaluation (TOE) used for the development of an ST for secure Windmill DMC Systems. Hence the description in this chapter is more specific and detailed in comparison with the PP TOE described in chapter 3. It is important to emphasise that the TOE model used for ST development is a constricted model of the PP TOE, in order to address a more specific type of TOE.

## 5.1 The TOE model

Referred to the general windmill DMC model defined in section 3.2, this section defines a more detailed Windmill DMC System model used as TOE for developing the ST. As such the ST TOE inherits the functionality from the PP TOE. The main difference between the two TOEs is in the architecture of the system. This is illustrated in figure 5.1.
As can be seen, the PP TOE is narrowed down into consisting of only one server which all entities in the TOE communicate via. The distributed feature is however still present since the entities in the TOE are physically distributed. The decision of having only one centralised server is made in order to simplify the model since with only one server no considerations are needed for specifying on

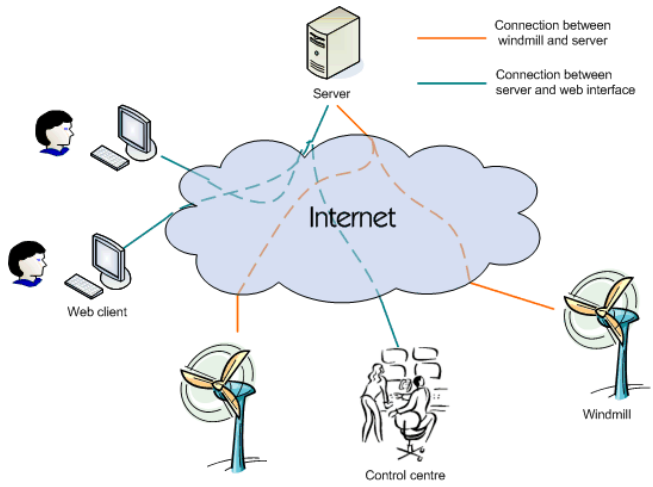


Figure 5.1: The overall structure of the TOE consisting of web interface, a web server, and windmills.

which servers and how data storage is located. Furthermore it is not needed to specify how servers interact. The communication is still done via the insecure Internet media.

## 5.2 TOE Components

The components that make up the TOE are described in this section. The roles and activity each component performs is presented and it is also considered and defined how the interaction between them should be.
Figure 5.2 shows how the components of the TOE are related and gives an overview of the data flows in the TOE.
The communication between a web client and the web server is based upon the HTTP protocol (Hyper Text Transfer Protocol). For the communication between the web server and the windmills the SOAP[1] protocol is used. By specifying which bare protocols the TOE makes use of, the ST TOE is restrained but still there is room for further specification. This could for instance be specification of type of messaging patterns in SOAP, or HTTP using Secure Socket Layer (SSL) encryption. The final specification can be done in the im-

[1]SOAP (Simple Object Access Protocol) is a simple XML-based protocol to let applications exchange information over HTTP.

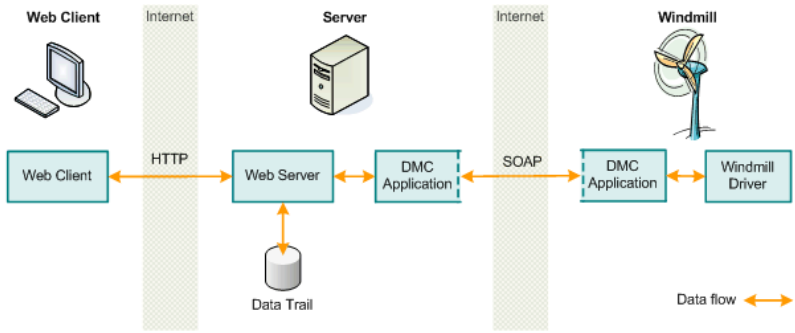


Figure 5.2: A schematic overview of the TOE.

plementation representation phase of the product development.
The reason for choosing a HTTP-SOAP solution is that these protocols have gained wider acceptance in constructing web services, as they work well with today's Internet infrastructure compared with other solutions [11], such as SMTP or FTP in stead of the HTTP protocol and CORBA, GIOP or ICE in stead of the SOAP protocol.
Below, the components/entities of the TOE are listed and described shortly:

**Web client**

The web interface is accessible through a web browser that supports HTTP communication. It provides a corporate login facility that is used to authenticate the user. The authentication is done by the web server before the user is allowed access to the TOE.
Upon successful authentication users are allowed to get an overview of the status of windmills. Furthermore, the web interface provides functions for changing windmill attributes.

**Server**

The server acts as a web server to the users that make use of the web interface to control and monitor windmills. Every time a user or another entity tries to gain access via the web interface the server registers the attempt into the data trail. The attempt is registered by noting when, from where, and by whom the attempt has been carried out.
As shown in figure 5.2 users can after login send requests concerning monitoring and controlling windmills to the server, whereafter the server registers the requests into the data trail and subsequently passes on the requests to the DMC application which takes of the web server-windmill communication by implementing SOAP. The DMC application is divided into two: the server-side DMC application (running on the server) and the

windmill-side DMC application (running on the windmill OS). The server-side of the DMC application transmits the requests to the windmill-side of the DMC application running on the windmills that the requests concern. The windmill-side of the DMC application then processes the requests by among other things making the Windmill Driver to act according to the commands of the requests.
After having processed the requests the DMC application sends back the replies. The web server registers the replies as responses to the corresponding received requests and then responds back to the web client.

The data trail, which the web server interacts with, is a database that stores event data which date minimum a month back. This includes audit data and windmill status data.
If a user wants to see status of a windmill dating back to for instance 2 weeks ago the server looks up the data in the data trail and sends back the status of the windmill in question for the requested date. But if a user wants to see the current status of a windmill, the server sends a request to the windmill and the corresponding reply is sent back.
In order to have a registry of the behaviour of the windmills over a period of time, it is necessary to read windmill data periodically. This is done by the web server sending status requests regularly to windmills through the DMC application.

**Windmill**
A windmill in the TOE, besides hardware components, consists of the windmill-side of the DMC application and a Windmill Driver which are running on the windmill operating system. The DMC application takes care of receiving requests sent from users through the server to the windmills in question and vice versa. The DMC application processes received data from both the web server and the Windmill Driver such that it is represented and passed on correctly (e.g. right format of data).
The Windmill Driver collects data about the windmill and furthermore can on basis of the requests give instructions to change windmill attributes (such as rotor speed, gear, brakes, produced amount of electricity, etc.), ie. the Windmill Driver is the application that is in connection with the hardware related parts of the windmill and can control and read data from these.

## 5.3 The Data Flows in the TOE

Since data is sent forth and back between TOE components via the insecure Internet media it is very relevant to analyse what kind of data is going to be

transmitted. The communication between TOE components is seen as a web client-web server and a web server-windmill interaction (see figure 5.2).
Firstly, the web client-web server communication is discussed. The type of data that flows between the web client and the web server can be divided into 2 groups: Requests and replies. The requests can furthermore be grouped into requests concerning user commands made by operations engineers and service technicians, and requests concerning administrative work done by administrators of the TOE. Replies to the requests can correspondingly be divided into 2 groups. Additionally, the web clients can fetch information about the state of the windmills so the shown information on the web interface is constantly updated.
Next, the web server-windmill interaction is considered. The web server can on behalf of users or on own initiative ask windmills for their status in order to reply back to users or keep the data trail up to date.
On the basis of these considerations the data flows in the TOE can be specified as below:

**User command requests**

Users that are allowed access to the TOE via the web interface can send requests concerning:

- Change of windmill attributes (allowed for operation engineers and administrators);
- View windmill attributes (allowed for all roles).

The web interface provides functionalities that allow users to make these requests. Furthermore, the web client can by itself send a "view" command to the web server in order to update the data on the screen with the newest information about windmills.

**User command replies**

As reply to user command requests the web server can, after processing the requests as described in 5.2, respond with the following:

- Messages about the status of made requests[2];
- Windmill attributes.

The responses are received by the web client and shown on the web interface.

**Administrative command requests**

Since the administrators have the responsibility of administrating the

[2]The status information could for instance be "Pending", "Changed", "Failed", or "Cancelled". There could be some additional information that can describe the status further more.

TOE, it is possible for them to make requests concerning view or change of data in the web server or the windmills.
The administrators are hence able to fetch log information by requesting the data trail from the web server. The administrators could be interested in for instance reading who and when entities have tried to access services provided by the web server. This could be done in context with auditing of the server in order to carry out required security procedures in the TOE. Additionaly, administrators can make requests concerning role management issues, e.g. adding/removing users, changing/updating user information, and changing/updating user rights and privileges in the TOE.

**Administrative command replies**
Replies to the administrator requests include information about whether the requests are carried out successfully or not. Additionally the replies can contain requested data.

**Windmill status messages**
These messages consist of data concerning windmill operation. This could be a message from the server to a windmill asking for status or a message with actual windmill data from windmill to server.

## 5.4 TOE Devices and Roles

In this section a brief overview of the roles and devices that can access the TOE will be given.

### 5.4.1 TOE Devices

The devices that are used to access the TOE are computers and laptops. This has been decided because computers and laptops are common work stations for employees in corporate organisations. The computers and laptops used in the TOE are devices provided by the organisation, and which are set up with required configurations by administrators in order to obtain a controlled environment from which access to the TOE is established. If a user makes use of a computer that administrators do not have control of, it is possible that the computer is infected with various malware which could jeopardise the security of the TOE.

### 5.4.2 TOE Roles

The roles in the ST TOE are the same as stated in the PP TOE description (see section 3.4.2). Since promotion/degradation, employment/dismissal, and change in range of responsibilities among roles can occur frequently in this kind of systems, the roles of the TOE are dynamic, ie. the rights and privileges of a role can change over time.

Chapter 6

# Security Target (ST)

In this chapter the development of the Security Target (ST) for the Windmill DMC System will be presented. The steps involved in developing a ST were previously described in chapter 2. A summary of the steps can be viewed in figure 6.1. The contents of this chapter should be read in parallel with appendix B.

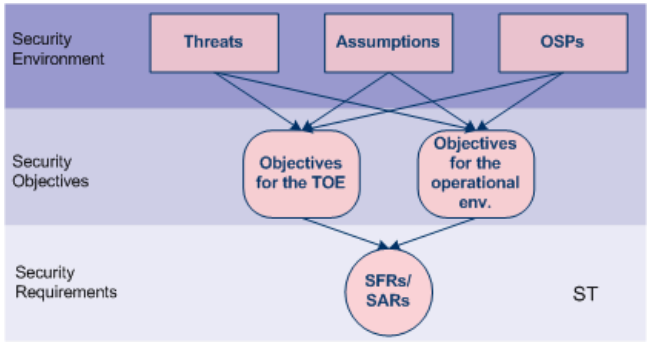


Figure 6.1: The ST process.

This ST claims conformance to the previously developed "Windmill Distributed Monitoring and Control System CC Protection Profile".

## 6.1 TOE Security Environment

In this section the TOE security environment will be discussed. This corresponds to the upper layer of the development phase of a ST (see figure 6.1). This includes identification of assets, threat agents, threats, secure assumptions, and any organisational policies (OSPs) that must be stated. Rather than give a repetition of the security environment stated in the PP, this section will focus on any addition or enhancement made to the security environment when a more specific and concrete TOE model is considered.

| Component | Modification |
|---|---|
| T.MASQUERADE | None |
| T.UNAUTHORISED_ACCESS | None |
| T.MODIFICATION | None |
| T.UNATTENDED_SESSION | None |
| T.ACCIDENTAL_USER_ERROR | None |
| T.DATA_TRANSMISSION | None |
| T.CRYPTO_LEAK | None |
| A.CORRECT_DEVELOPMENT | None |
| A.NO_EVIL | None |
| A.PHYSICAL | None |
| A.EXTERNAL_PARTY | Added |
| P.AUTHORISED_USERS | None |
| P.USER_PRIVILEGES | None |
| P.ACCOUNTABILITY | None |
| P.CRYPTOGRAPHY | Enhanced |
| P.TRAIN | None |

Table 6.1: The security environment modifications.

In table 6.1 the threats, assumptions, and OSPs that constitute the TOE security environment are listed. Notice that additions to or enhancements of the security environment are stated in the table too.

### 6.1.1 Threats to Security

In this section the assets, threat agents, and the threats against the security of the TOE will be described and discussed relative to the PP.

#### 6.1.1.1 Assets and Threat Agents

The assets of the ST TOE, that need to be protected, do not change from the assets of the PP TOE, since they are derived directly from the purpose of the TOE. This means that the assets are:

- The data trail, and
- monitoring and control data.

The data trail includes audit data and status data of windmills.

Threat agents are the same as defined in the PP.

#### 6.1.1.2 Threats

When considering the threats on ST level as opposed to the PP, there has not been identified any changes.

### 6.1.2 Secure Usage Assumptions

One new assumption has been added to the previous stated assumptions in the PP. The reason for including this new A.EXTERNAL_PARTY assumption is because when observing the specific TOE it is necessary to assume that any external products that the TOE will have to rely upon when upholding its security are trusted. For instance operating systems, cryptographic services, and access control mechanisms that will be provided by external products have to be trusted and consider as safe enough for being applied in the TOE.

**A.EXTERNAL_PARTY** ◇ *Any external parties and products (operating systems, cryptographic services, access control mechanisms, etc.) which the TOE relies upon are assumed trusted and fully functioning.*

Thereby it is assumed that external products are trusted and correctly installed and configured.

### 6.1.3 Organisational Security Policies (OSPs)

Any modifications made to the OSPs will be discussed in this section. From table 6.1 it was stated that only the policy P.CRYPTOGRAPHY has undergone enhancement.

**P.CRYPTOGRAPHY** ◇ *All cryptographic services used in the TOE must comply with the Federal Information Processing Standard Publication (FIPS PUB) 140-2 level 1.*

This policy is enhanced such that it is specified which standard to use when applying cryptographic modules in the TOE. A cryptographic module is that part of a system or application that provides cryptographic services, such as encryption, authentication, or electronic signature generation and verification. Products and systems compliant with this ST are expected to apply cryptographic modules compliant with FIPS 140-2 level 1 [30].

## 6.2 Security Objectives

With identification of new or enhanced assumptions and OSPs the next step in the ST development phase is to specify the security objectives for both the TOE and for the operational environment. The security objectives defined in the PP must now be revised in order to determine if they are still valid for the ST security environment.
Table 6.2 shows an overview of the objectives stated in the ST. Furthermore, the table shows whether the objectives are modified or not. As can be seen it is found that actually only the O.CRYPTO_FUNCTIONS objective has to be enhanced. The enhancement is described as follows:

**O.CRYPTO_FUNCTIONS** ◇ *The TSF shall implement functions that comply with the* ***Federal Information Processing Standard Publication (FIPS PUB) 140-2 level 1****.*

This security objective states that proper cryptographic measures shall be provided in securing the TOE. This includes securing data during transmission. Any cryptographic function in the TOE, ie. encryption/decryption, authentication, and signature generation/verification, and key generation, is dealt with by this objective. Therefore the objective counters T.MODIFICATION and T.DATA_TRANSMISSION, and ensures

| Objective | Modification |
|---|---|
| O.UNIQUE_IA | None |
| O.DATA_INTEGRITY | None |
| O.ACCOUNTABILITY_AND_AUDIT | None |
| O.CRYPTO_FUNCTIONS | Enhanced |
| O.ROLE_MANAGEMENT | None |
| O.SESSION | None |
| O.BACK-UP | None |
| O.VULNERABILITY_ANALYSIS | None |
| O.CRYPTO_SECRECY | None |
| O.SELF_TEST | None |
| OE.TRAIN | None |
| OE.ISOLATION | None |

Table 6.2: Overview of objectives in the ST.

P.CRYPTOGRAPHY. The TSF shall follow the FIPS PUB 140-2 level 1 in order to implement this objective.

It has to be kept in mind that the general environment has not changed, and thus the security objectives for the environment are left unchanged.

The enhanced or added assumptions and policies, and the way in which these are met by the objectives, are as follows:

**A.EXTERNAL_PARTY** ⋄ The new assumption stated in the security environment, enables the TOE to trust external parties and products. When applying products developed by external parties it is determined that the products must as minimum live up to the security requirements of the TOE itself. This means objectives for unique identification and authentication, data integrity, cryptographic functions being FIPS validated, and any cryptographic keys being kept secure address this assumption and enforces the external parties and products to meet these objectives too.
Proper installations and configurations are the job of administrators of the TOE. So OE.TRAIN also addresses the assumption in order to verify that administrators are reliable to do their work properly.
O.SELF_TEST also has an impact in realising the assumption, since a self test could detect flaws in the products and enforces appropriate measures to be taken.

**P.CRYPTOGRAPHY** ⋄ This policy was enhanced in the sense that it had to ensure that specifications in FIPS 140-2 level 1 are followed when applying

cryptographic services. With the enhancement of the objective for cryptographic functions, discussed earlier in this section (O.CRYPTO_FUNCTIONS), this policy is covered. Furthermore, secure appliance of cryptography is also ensured by O.CRYPTO_SECRECY, as discussed in the PP (see section 4.2). Again a last resort solution to uphold this policy is given by O.SELF_TEST, since it can detect flaws in the cryptographic modules used by the TSF.

Summarised, table 6.3 shows the mapping of objectives to threats, OSPs, and assumptions. Notice that, just like in the PP, objectives that partly have an impact on threats, assumptions, and/or OSPs are marked with a (X). These markings will not appear in the actual ST.

| | O.UNIQUE_IA | O.DATA_INTEGRITY | O.ACCOUNTABILITY_AND_AUDIT | O.CRYPTO_FUNCTIONS | O.ROLE_MANAGEMENT | O.SESSION | O.BACK-UP | O.VULNERABILITY_ANALYSIS | O.CRYPTO_SECRECY | O.SELF_TEST | OE.TRAIN | OE.ISOLATION |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T.MASQUERADE | × | × | × | | | | | | | (×) | | |
| T.UNAUTHORISED_ACCESS | × | × | × | | × | | | | | (×) | | |
| T.MODIFICATION | × | × | × | × | | | × | | | (×) | | |
| T.UNATTENDED_SESSION | | | | | | × | | | | (×) | × | (×) |
| T.ACCIDENTAL_USER_ERROR | | | × | | | | × | | | (×) | × | |
| T.DATA_TRAMSMISSION | | × | | × | | | | | | (×) | | |
| T.CRYPTO_LEAK | | | | | | | | | × | (×) | | |
| A.CORRECT_DEVELOPMENT | | | | | | | | × | | (×) | | |
| A.EXTERNAL_PARTY | × | × | | × | | | | | × | (×) | × | |
| A.NO_EVIL | | | | | | | | | | (×) | × | |
| A.PHYSICAL | | | | | | | × | | | (×) | | × |
| P.AUTHORISED_USERS | × | × | × | | × | | | | | (×) | | |
| P.USER_PRIVILEGES | | | | | × | | | | | (×) | | |
| P.ACCOUNTABILITY | × | | × | | × | | | | | (×) | | |
| P.CRYPTOGRAPHY | | | | × | | | | | × | (×) | | |
| P.TRAIN | | | | | | | | | | (×) | × | |

Table 6.3: Objectives related to the TOE security environment.

## 6.3 Security Requirements

This section introduces the requirements needed to fulfill security objectives for the ST TOE. This includes specification of SFRs and SARs (through EAL selection). This corresponds to the lower layer of the ST development (see figure 6.1). Since the ST claims conformance to the 'Windmill Distributed Monitoring and Control System CC Protection Profile', the requirements stated in this section will focus on any modifications and enhancements of these relative to the PP.

### 6.3.1 Security Functional Requirements (SFRs)

In this section SFRs for the ST TOE, which have undergone selection or value assignment, will be discussed. Specification of values to selections and assignments in the identified components will be written in *italic* text.

#### 6.3.1.1 Unique Identification and Authentication

The SFRs that ensure the O.UNIQUE_IA objective, and that have been made selections for and assigned values to in the ST, are described below.

***FIA_UAU.3 - Unforgeable authentication***
In this component the following question has to be considered: Shall the TSF **detect** or **prevent forged** and **copied** authentication data? Since the intention is to achieve a proactive security system, it is clear that *prevention* is appropriate. It is noticed that the CC looks at passwords as not-controllable by the TSF in contrast to biometric authentication solutions, since passwords can be passed on to other entities either voluntarily or by theft. Never the less biometrics can in extreme cases, where an owner of authentication data voluntarily or by force has to give the information to unauthorised entities[1], not be controlled by the TSF. But these cases are not considered in this project since they are too extreme.

***FIA_UAU.6 - Re-authenticating***
Re-authentication requires clear specification of the conditions under which re-authentication is required by the TSF. *The TSF shall re-authenticate users*

[1]Say fx. if authentication is done by iris recognition and an attacker cuts out your eye. Sorry for sounding violent.

*when they want to unlock a session.* This is the only condition under which this component is applicable.

#### 6.3.1.2 Accountability and Audit Records

The components addressing accountability and audit records are discussed in the following. Selections and assignments of values related to the components are presented.

***FAU_ARP.1 - Security alarms***
When intrusion or violation of TOE security functions has been detected, the TSF has to take one or more of following actions in order to stop the intrusion or correct the security violation:

*a) Show the alert visually;*

*b) Send e-mail to administrators;*

*c) Apply Simple Network Management Protocol (SNMP) Traps*[21]*;*

*d) Block traffic.*

Which of these actions the TSF shall take depends on which kind of security violation that is in question.
SNMP traps enable an agent to give notification of significant events by way of an unsolicited SNMP message [21]. This, together with emails, shall be used in order to alarm administrators.
Sometimes it is useful to deny data flow when an intrusion or other violation of security has occured. This could be done by blocking traffic to/from the TOE (by using for instance firewalls).

***FAU_SSA.1 - Potential violation analysis***
With this SFR a set of rules in monitoring the audited events, is specified, which then will be used by the TSF to indicate a potential violation and trigger any security alarms stated in FAU_ARP.1. The rules for monitoring audited events are as stated:

- Accumulation or combination of *all in FAU_GEN.1 specified auditable events*, known to indicate a potential security violation;
- *Start-up and shutdown of the audit functions.*

***FAU_GEN.1 - Audit data generation***
In this component it will be specified which auditable events should be generated and recorded as audit by the TSF. The level of audit is chosen to be the *minimum* level. This is due to considerations made about costs and resources (see section 4.1.4). The auditable events are:

- Start-up and shutdown of the audit functions;
- *All auditable events specified in table 6.4;*[2]
- *Start-up and shutdown of the server and windmill applications; and*
- *all user-initiated events.*

The information that should be recorded are date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event. Furthermore the TSF shall record the *IP and MAC addresses* of entities (e.g. users via web clients and windmills) communicating with the server.

***FAU_SAR.1 - Audit review***
Since audit records are evidence of the activity within the TOE and can be used to detect unwanted activity it shall be possible for *administrators* to read *all audit information* from the audit records.

***FAU_STG.1 - Protected audit trail storage***
Since the TSF has to react proactive, it is appropriate to *prevent* (in stead of only detecting) unauthorised modifications to the stored audit records in the audit trail.

#### 6.3.1.3 Management

Security functional requirements regarding management are discussed in this section. This includes specification of the management functions and security functions in particular.

***FMT_MOF.1 - Management of security functions behaviour***
This component requires selection of appropriate management features of stated

[2]Table 6.4 contains components that can be audited on the minimal level of audit. Descriptions of what has to be recorded as audit is defined in the CC and is reproduced in this table.

| Functional Component | Auditable Event |
|---|---|
| FAU_ARP.1 | Actions taken due to potential security violations. |
| FAU_SAA.1 | Enabling and disabling of any of the analysis mechanisms. |
| | Automated responses performed by the tool. |
| FCS_CKM.1<br>FCS_CKM.2<br>FCS_CKM.4 | Success and failure of the activity. |
| FCS_COP.1 | Success and failure, and the type of cryptographic operation. |
| FDP_IFF.1 | Decisions to permit requested information flows. |
| FDP_ITT.1 | Successful transfers of user data, including identification of the protection method used. |
| FDP_SDI.1 | Successful attempts to check the integrity of user data, including an indication of the results of the check. |
| FIA_UAU.2 | Unsuccessful use of the authentication mechanism |
| FIA_UAU.3 | Detection of fraudulent authentication data. |
| FIA_UAU.6 | Failure of reauthentication. |
| FIA_UID.2 | Unsuccessful use of the user identification mechanism, including the user identity provided. |
| FMT_MSA.2 | All offered and rejected values for a security attribute. |
| FMT_SMF.1 | Use of the management functions. |
| FMT_SMR.1 | Modifications to the group of users that are part of a role. |
| FPT_ITT.1 | The detection of modification of TSF data. |
| FPT_RCV.1 | The fact that a failure or service discontinuity occurred. |
| | Resumption of the regular operation. |
| FPT_STM.1 | Changes to the time. |
| FTA_SSL.1 | Locking of an interactive session by the session locking mechanism. |
| FTA_SSL.2 | Successful unlocking of an interactive session. |

Table 6.4: Additional auditable events for the minimal level of audit, from CC components.

security functions within the TSF. Which roles that are to be assigned to manage these functions are specified too by this component. It should be restricted by the TSF that only *administrators* are able to *determine the behaviour* and *modify the behaviour* of the functions. It should not be possible to switch on/off the security functions since disabling a function leaves the TOE vulnerable and this should not be the case. The security functions determined are as stated in section 4.3.1.3.

***FMT_MSA.1 - Management of security attributes***
The *DATA_FLOW_SFP* (see appendix B section B.7.2) is enforced by the TSF through this component. It restricts the ability to *modify* the security attributes *that are defined in the SFP* to *administrators*.
The ability to only modify[3] security attributes has been selected since it is found that in regards to maintenance of security attributes this is sufficient.

***FMT_MSA.3 - Static attribute initialisation***
The TSF shall enforce the *DATA_FLOW_SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP. Restrictive default values for attributes are chosen since security has to be relatively high. It might occur, that even though the default values can be changed, it actually may not happen because it simply has been forgotten that they can/should be changed. Only *administrators* are allowed to specify alternative initial values to override the default values when an object or information is created.

***FMT_MTD.1 - Management of TSF data***
In this component it is specified that only *administrators* are allowed to manage following TSF data:

*a) Data trail;*

*b) Identification and authentication data;*

*c) Cryptographic algorithms and keys;*

*d) Audit records.*

The operations that are included in the management of the TSF data in this component are: *Modify*, *delete*, and *query* operations. These operations are found sufficient in order to manage TSF data. For instance, administrators have to be able to add or remove user accounts or change/update cryptographic procedures, and for this the identified operations are sufficient.

[3]Could select among the operations change_default, query, modify, delete, or another operation specified by the ST developers.

#### 6.3.1.4 Role Management

Relative to the PP, there are no changes in components addressing this topic. The component "FMT_SMR.1 - Security roles" still ensures management of roles.

#### 6.3.1.5 Session

Two components were identified in the PP to address session locking, namely "FTA_SSL.1 - TSF-initiated session locking" and "FTA_SSL.2 - User-initiated locking".

***FTA_SSL.1 - TSF-initiated session locking***
In this component it is required to assign a time interval of inactivity before the TSF is to lock a session. In order to decide this, it is important to take into consideration that the time interval should not be an irritation to users. It is found suitable that *15 minutes* inactivity is reasonable.
Furthemore, it should be specified which events shall occur prior to the TSF unlocking the session. This is closely related to the FIA_UAU.6 component for *re-authenticating* (see section 6.3.1.1), since this is the only event required to unlock a session.

***FTA_SSL.2 - User-initiated locking***
Session locking initiated by users does not require any inactivity time interval, but still requires specification of events that shall occur before the TSF shall unlock the session. As with the previous session locking component, the only event that can lead to unlocking is *re-authentication*, ie. FIA_UAU.6.

#### 6.3.1.6 Cryptography

Again, no additional security components, beyond those stated in the PP, are required to meet the cryptographic functionality of the TOE. At ST level it is though required to specify which concrete methods and algorithms for cryptographic operations are to be implemented. This will be done in the following for each of the stated cryptographic components. The component specification comply with the FIPS 140-2 level 1 standard [30].

***FCS_COP.1 - Cryptographic operation***
In order to achieve a secure communication between the TOE components (ie. web client, server, and windmills) over the Internet it is necessary to specify

which operations the TOE shall support. This includes encryption/decryption of data and mutual authentication among the TOE components.
The Secure Socket Layer (SSL) protocol or the Transport Layer Security (TLS) protocol shall be used to implement these cryptographic operations.
The SSL and TLS protocols allow client/server applications to communicate in a way designed to prevent, among other things, eavesdropping, tampering, and message forgery [11].
TLS v1.0 actually defines the same protocol as SSL v3.1. This means that the TLS cipher suites[4] that provide the cryptographic operations, can be used:

*a) TLS_RSA_WITH_AES_128_CBC_SHA*

*b) TLS_RSA_WITH_AES_256_CBC_SHA*

As can be seen from the listed cipher suites, they consist of an asymmetric and a symmetric algorithm. The asymmetric algorithm (RSA) is used to distribute cryptographic keys and thereby account for authentication. The RSA algorithm is one of the most used asymmetric algorithms. Its use for distributing symmetric keys is very common, since it is a slow but safe algorithm which is most effective when using relatively short pieces of data (like keys). The key size that should be used for RSA public key encryption has to be 1024 bit long since this key length is recommended by the RSA Security Inc. for corporate use [13].
The symmetric algorithm (AES) is used for the actual encryption/decryption of data. The AES is found appropriate since this algorithm is one of the most popular and in comparison with other symmetric algorithms (e.g. DES or RC4) it is much faster, easier to implement, and requires less memory [11]. The key sizes for the AES are defined by choice of cipher suite, ie. it can either be *128 bits* or *256 bits*.

***FCS_CKM.1 - Cryptographic key generation***
This security component deals with the issue of generating cryptographic keys. The algorithm for key generation is specified to be a *Secure Hash Algorithm (SHA) as stated in the Secure Hash Standard (SHS)* [31]. Exactly which SHA to choose is dependent on the length of output that is desired, ie. key length. The key length size the SHA must provide, is *dependent on which actual cipher suite is selected for the TOE to base its cryptographic operations upon.* These algorithms are described under FCS_COP.1. Furthermore, the SHA algorithm is specified to use random number generation when producing output for keys. This is done because it should be impossible to reproduce the key for other parties. The reason for choosing a secure hash algorithm as key generation algorithm is, that it is widely used in the real world and its secure deployment in

[4]The cipher suites are described in [20] and [24]

several applications is well known [29].

***FCS_CKM.2 - Cryptographic key distribution***
The cipher suites both require keys distributed by *the RSA key exchange algorithm*. Public key infrastructure (PKI) is used in order to obtain mutual authentication between the TOE components when exchanging keys.

***FCS_CKM.4 - Cryptographic key destruction***
In order to destroy keys appropriately, *procedures described in the FIPS 140-2 level 1* shall be followed.

#### 6.3.1.7 Data Protection

Two types of data need careful consideration for protection: User data and TSF data. Value selection and assignment for components aiming at data protection are discussed in this section.

**User Data Protection**

***FDP_IFC.1 - Subset information flow control***
By this component the TSF enforces the *DATA_FLOW_SFP* on *data flow* between *the web clients and the web server*, and *the web server and the windmills*. The SFP is explained in section B.7.2.

***FDP_IFF.1 - Simple security attributes***
The TSF shall enforce the *DATA_FLOW_SFP* specified in FDP_IFC.1 based on types of subject and information security attributes. The list of subjects and information controlled under the indicated SFP, and for each, the security attributes are specified in this SFR:

**Subject security attributes:**

*a) Type of input/output devices which the information flows between;*

*b) Roles of the entities that cause the information to flow or act as recipients of the information.*

**Information security attributes:**

*a) Type of information;*

*b) Sensitivity of the information.*

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

*a) The input/output devices are authenticated before any data flow is allowed between them;*

*b) Entities own the rights to carry out actions that imply data flow (e.g. a service technician is not able to create a data flow for changing windmill attributes);*

*c) Data is encrypted using one of the cipher suites TLS_RSA_WITH_AES_128_CBC_SHA or TLS_RSA_WITH_AES_256_CBC_SHA in SSL 3.1/TLS 1.0 before data flow is allowed between TOE components.*

***FDP_ITT.1 - Basic internal transfer protection***
This SFR ensures protection of user data when it is transferred within the TOE. For this the TSF shall enforce the *DATA_FLOW_SFP* specified in appendix B section B.7.2 to prevent *disclosure* and *modification* of user data.

***FDP_SDI.1 - Stored data integrity monitoring***
This component is included in order to ensure protection of stored data by monitoring user data stored in containers controlled by the TSF for specified integrity errors on all objects, based on defined user data attributes. The CC states the following about the FDP_SDI family:

> *To prevent a subject from modifying the data, the Information flow control functions (FDP_IFF) or Access control functions (FDP_ACF) families are required (rather than this family) ([5] section F.11).*

Therefore this component is solely included because of data integrity violations that may be caused by hardware glitches or errors. Thus this SFR is not meant to cover prevention of data modification by subjects (this is dealt with in the FDP_IFF component). The integrity errors that the TSF shall monitor for are:

*a)* Data that is changed or lost because of errors in the hardware.

The monitoring of user data shall be based on following user data attributes:

*a) The contents of the data;*

*b) The creator of the data;*

*c) Date of creation and modification of the data;*

*d) Read/write permissions of data.*

The mechanism for checking data integrity can be carried out by fx. CRC or taking a "snap shot" of data regularly and during an integrity check, the current state of data should be compared to the snap shot. The mechanism could furthermore provide procedures for informing administrators when a violation or any illegal change is detected.

**TSF Data Protection**

The class of CC that addresses this area is FPT - Protection of the TSF. This class specifies SFRs that ensure integrity and management of TSF functionalities and integrity of TSF data.

***FPT_AMT.1 - Underlying abstract machine test***
As mentioned in the PP, the underlying abstract machine is a virtual or physical machine upon which the TSF executes. In order to verify the security assumptions, such as memory capacity and correct mode of operation, made about the underlying abstract machine the TSF shall run a suite of tests during following conditions:

*a) During initial start-up;*

*b) Periodically during normal operation;*

*c) At the request of an authorised user, ie. an administrator.*

Since the TOE has to be running constantly, it is obvious that tests should occur periodically and whenever requested by an administrator.

***FPT_ITT.1 - Basic internal TSF data transfer protection***
With this component the TSF data shall be protected from *disclosure* and *modification* when it is being transferred between physically-separated parts of the TOE via internal channels.

#### 6.3.1.8 Self Test

Another component that ensures preservation of security of the TOE is the FPT_TST.1 component. The assignments and selections made for this component are described below.

***FPT_TST.1 - TSF self test***
This component specifies conditions under which self test should occur and the integrity of which parts of the TSF should be verified. The TSF shall run a suite of self tests:

*a) During initial start-up;*

*b) Periodically during normal operation;*

*c) At the request of an authorised user, ie. administrator.*

The tests shall demonstrate the correct operation of *the TSF*.

#### 6.3.1.9 Back-up

In order to carry out appropriate back-up in the TOE, following SFRs with the specified selections and value assignments are needed:

***FPT_RCV.2 - Automated recovery***
In this component it is selected that the TSF shall enter a maintenance mode where it is possible to return to a secure state when *power failure* and *system failures*, occur.
Furthermore, when power failure occurs, the TSF shall by using automated procedures, ensure the return of the TOE to a secure state, ie. this shall be done without human intervention. In the case of system failure, the TOE will require re-booting.

### 6.3.2 Evaluation Assurance Level (EAL) Selection

The selection of EAL corresponds to the selection of security assurance requirements (SARs). This section describes this selection of EAL for the ST. In the PP case the *EAL3 - methodically tested and checked* was found adequate.

Even though the TOE has been narrowed down to a more specific TOE, the general threat scenario is still the same. Thus EAL3 is still found suitable in assuring the security of the TOE. There is not added any further SFRs, which means that no extra dependencies are present, so EAL3 is still applicable. The included assurance requirements are listed in table 6.5. How these assurance re-

| *Assurance Class* | *Assurance components/SARs* | |
|---|---|---|
| ADV:Development | ADV_ARC.1 | Security architecture description |
| | ADV_FSP.3 | Functional specification with complete summary |
| | ADV_TDS.2 | Architectural design |
| AGD:Guidance documents | AGD_OPE.1 | Operational user guidance |
| | AGD_PRE.1 | Preparative procedures |
| ALC:Life-cycle support | ALC_CMC.3 | Authorisation controls |
| | ALC_CMS.3 | Implementation representation CM coverage |
| | ALC_DEL.1 | Delivery procedures |
| | ALC_DVS.1 | Identification of security measures |
| | ALC_LCD.1 | Developer defined life-cycle model |
| ASE:Security Target evaluation | ASE_CCL.1 | Conformance claims |
| | ASE_ECD.1 | Extended components definition |
| | ASE_INT.1 | ST introduction |
| | ASE_OBJ.2 | Security objectives |
| | ASE_REQ.2 | Derived security requirements |
| | ASE_SPD.1 | Security problem definition |
| | ASE_TSS.1 | TOE summary specification |
| ATE:Tests | ATE_COV.2 | Analysis of coverage |
| | ATE_DPT.1 | Testing: basic design |
| | ATE_FUN.1 | Functional testing |
| | ATE_IND.2 | Independent testing - sample |
| AVA:Vulnerability assessment | AVA_VAN.2 | Vulnerability analysis |

Table 6.5: Security assurance components in EAL3 [7].

quirement are met, is discussed in the assurance measures in appendix B section B.7.3.

## 6.4 TOE Summary Specification

The TOE summary specification outlines the security functions, security functional policies, and assurance measures of the TOE that meet the TOE security requirements. These are specified in appendix B in section B.7.
The security functions reflect the actual security functionality of the TOE and are identified on basis of the TOE architecture, security objectives, and security functional requirements for the TOE. The identified security functions can be used as basis for developing possible implementation representations. The mapping of security functions to security functional requirements is shown in table 6.6. The definitions of the security functions can be read in appendix B section B.7.1.
The assurance measures are descriptions of how the assurance requirements are met.

| | F.BACK-UP | F.ROLE | F.AUDIT | F.AUTH | F.CRYPTOGRAPHY | F.SCAN | F.MANAGEMENT |
|---|---|---|---|---|---|---|---|
| FAU_ARP.1 | | | × | | | | |
| FAU_GEN.1 | | | × | | | | |
| FAU_GEN.2 | | | × | | | | |
| FAU_SAA.1 | | | × | | | | |
| FAU_SAR.1 | | | × | | | | |
| FAU_SAR.2 | | | × | | | | |
| FAU_STG.1 | | | × | | | | |
| FCS_CKM.1 | | | | | × | | |
| FCS_CKM.2 | | | | | × | | |
| FCS_CKM.4 | | | | | × | | |
| FCS_COP.1 | | | | | × | | |
| FDP_IFC.1 | | | | × | × | | |
| FDP_IFF.1 | | | | × | × | | |
| FDP_ITT.1 | | | | × | × | | |
| FDP_SDI.1 | | | × | | | × | |
| FIA_UAU.2 | | | | × | | | |
| FIA_UAU.3 | | | | × | | | |
| FIA_UAU.6 | | | | × | | | |
| FIA_UID.2 | | | | × | | | |
| FMT_MOF.1 | | | | | | | × |
| FMT_MSA.1 | | | | | | | × |
| FMT_MSA.2 | | | | | | | × |
| FMT_MSA.3 | | | | | | | × |
| FMT_MTD.1 | | | | | | | × |
| FMT_SMF.1 | | | | | | | × |
| FMT_SMR.1 | | × | | | | | |
| FPT_AMT.1 | | | | | | × | |
| FPT_ITT.1 | | | | × | × | | |
| FPT_RCV.2 | × | | | | | | |
| FPT_STM.1 | | | × | | | | |
| FPT_TST.1 | | | | | | × | |
| FTA_SSL.1 | | | | × | | | |
| FTA_SSL.2 | | | | × | | | |

Table 6.6: Mapping of security functions to security functional requirements.

## 6.5 ST Conclusion and Comments

In this chapter a Security Target is developed for the TOE described in chapter 5. The EAL3 has been chosen as appropriate assurance level.

Where the PP is defined relatively broad, the ST specifies and restricts objectives and requirements. The ST TOE, relative to the PP TOE, is narrowed down to consisting of web clients, windmills, and only one web server. It should be kept in mind that access to the TOE is still done through a web interface which causes data flows between web clients and the web server. In order for the web server to reply back on requests from users, the server has to interact with windmills, which gives rise to data flows also between the web server and the windmills.

Even though the requirements in the ST have been constrained, developers have still been given the opportunity to develop several different implementation representations and concrete designs from the ST. For instance, the bare protocols HTTP and SOAP for data transmission have been specified in the ST but it is still left open for the developers to configure the actual data transfer and to decide which cryptographic solutions should be used.

The ST does not include any new threats, objectives, or policies but instead enhances P.CRYPTOGRAPHY and O.CRYPTO_FUNCTIONS by specifying a cryptographic standard that the TOE should follow. A new assumption A.EXTERNAL_PARTY is added in order to trust any external product used in the TOE, such as operating system, virus scanners, bare protocols, and cryptographic services.

It should also be noticed that the environment in which the TOE is to operate and thus objectives for the environment are not changed from the PP.

It is up to the developers of the ST to make appropriate and realistic value assignments and selections. This concludes that developing a ST is dependant on available resources and the level of security one wishes to achieve.

Chapter 7

# Implementation Representation/Design

This chapter will present an example of an implementation representation of a Windmill DMC System. The example will be used to specify how the security functions stated in the ST can be implemented in the context of a concrete application, so that the implementation representation meets the security functional requirements and the security assurance requirements for documentation and testing at EAL3 level of assurance, as specified in the ST.
Firstly, the example will be described in form of a design proposal, including an overview of the system architecture.
Then, the security functionality will be presented in form of a suggestion to how the security functions stated in the ST can be implemented in the design proposal. This includes identification of concrete components selected to implement the security functions (see figure 2.3).

## 7.1 Design of the TOE

The concrete application that will be used as basis for the rest of this chapter is presented and described in this section. This is in compliance with the requirements for documentation of the design as stated in EAL3 in the ADV

class, which defines requirements for description of the TOE development. This means the assurance measures M.ARCH and M.SPEC as described in the ST (appendix B section B.7.3) will be met in this section.

### 7.1.1 System Architecture

The system architecture is illustrated in figure 7.1. As can be seen the overall system architecture is divided into 3 parts:

- Web clients
- Web server
- Windmills

Furthermore the data flow between these 3 parts via the Internet media are shown. The protocols used for communication are the HTTPS protocol for the web client-web server communication, and the SOAP protocol for the web server-windmill communication (in compliance with the ST).
In the suggested architecture it is also tried to show the environment in which each part of the TOE operates and which have an impact on the physical security of the TOE. The web server is placed in a safety room, while windmills are gathered in windmill parks surrounded by security fences. In this way, it is tried to meet the assumption A.PHYSICAL stated in the ST. Notice that the environment of the web clients has not been depicted, since this is not controllable by the TSF.
The system architecture makes it possible for several web clients to make use of the web service and gain access to the services provided by the TOE. Since stand-alone windmills have come more and more uncommon due to maintenance costs, optimisation of power production and distribution, and security, only the scenario of windmill parks is reproduced in the example.

In the ST TOE description the general architecture was partly sketched (see section 5.1-5.3). In the following the architecture will be specified in concrete application.

#### 7.1.1.1 Web Clients-Web Server Architecture

The web server is implemented by the Oracle Application Server, which together with installation of a SSL/TLS certificate, provides HTTPS communication to

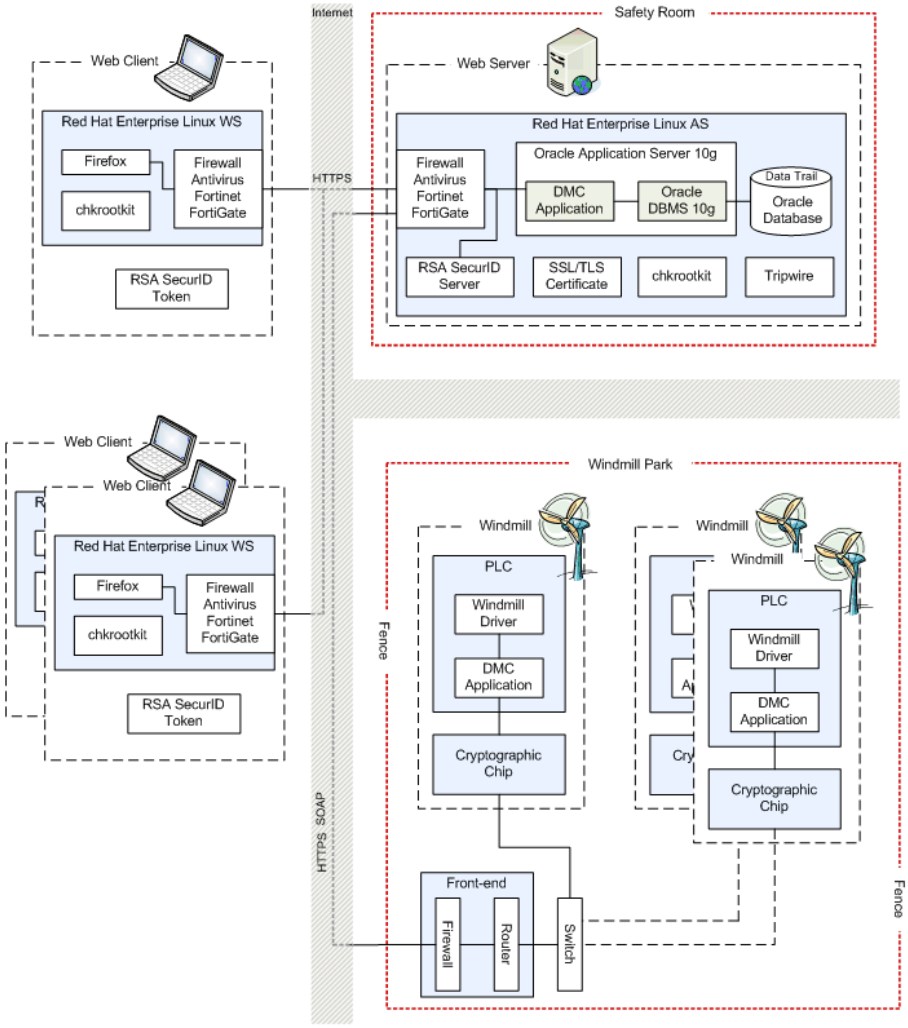


Figure 7.1: High-level design of the TOE.

web clients. The Oracle Application Server also implements the server-side DMC Application and the Oracle DBMS for storing the data trail and other application data. Together with a RSA SecurID Server, the Application Server implements access control to the web service.

#### 7.1.1.2 Web Server-Windmills Architecture

The server-side DMC Application acts as a client program to the windmill-side DMC Application which therefore is a small server program waiting for incoming requests from the web server.
Each of the windmill parks are equipped with a front-end which consists of a firewall and a router. The front-end provides control of traffic to/from the windmills. From the front-end a switch directs the data in a wired Ethernet to the respective windmills in the windmill park.
Since the communication between the web server and the windmills has to be done such that confidentiality and integrity of the data is maintained, the DMC Application on both sides are enforced the DATA_FLOW_SFP stated in section B.7.2 in the ST and by the FDP_IFC and FDP_IFF SFRs. How this is done is described in detail in the following sections.

## 7.2 Security Functionality

The functional specification of the TOE design is divided into sections that discuss each of the various parts of the TOE (ie. web clients, web server, and windmills). It should be noticed that the general functionality of the system (e.g. TOE model and data flows) is already explained in the description of the ST TOE (chapter 5) and therefore it is implied to this implementation representation too. Therefore this section should be read with the ST TOE as reference.

In the following sections, the considerations about which security functionalities and which concrete components that should be applied in each part of the TOE, will be discussed.
The concrete components are selected by considering how much costs and resources are affordable in order to obtain a security level corresponding to at least EAL3. The amount of money and other resources that shall be spent, have to be considered in relation to the severity of the consequences and loss, caused by security breaches, for the organisation. It should also be said that the components are enforced the A.EXTERNAL_PARTY assumption so that they are assumed trustworthy and reliable.

| | F.BACK-UP | F.ROLE | F.AUDIT | F.AUTH | F.CRYPTOGRAPHY | F.SCAN | F.MANAGEMENT |
|---|---|---|---|---|---|---|---|
| chkrootkit | | | | | | × | |
| DMC Application | | | | × | × | | × |
| Firefox 2 | | | | × | × | | × |
| Fortinet FortiGate Firewall Antivirus | | | | × | | × | × |
| Oracle Application Server 10g | | × | | | | | × |
| Oracle Database 10g | × | | × | | × | | |
| Oracle DBMS 10g | × | | × | × | × | | × |
| PLC | | | | | | | |
| Red Hat Enterprise Linux AS | × | × | × | × | | | × |
| Red Hat Enterprise Linux WS | | | | | | × | × |
| RSA SecurID Server | | | | × | × | | × |
| RSA SecurID Token | | | | × | × | | |
| TLS1.0 | | | | × | × | | |
| Tripwire | | | × | | | × | |
| Windmill Driver | | | | | | | |
| Cryptographic Chip | | | | × | × | | × |

Table 7.1: Mapping of concrete system components and security functions.

In cases where no evaluated products for a specific security function is available, it is tried to choose a product which seems to meet the security functionality to best effect, and which is of of high repute and widely used in the real world. Table 7.1 shows a mapping of selected components to security functions they satisfy.

### 7.2.1 Web Clients

The web clients can only be run on devices provided by the TOE, because it is wished to control from which devices it should be possible to access the TOE (see section 5.4.1). These devices run the operating system Red Hat Enterprise Linux WS Version 3 Update 2, which is EAL3+ evaluated [19].
The operating system is directly related to the security in a system, since it might have an impact on or directly implement identified security functions. Several factors have been taken into consideration when making this choice of

operating system for the devices running web clients. Below, the factors are listed in a prioritised order with highest priority stated first:

- Security
- Costs
- Usability

It is a fact that the most common operating systems are Windows and Linux OSs. Therefore, the options are narrowed down into consisting of only these two operating systems.
One of the reasons for using a Linux OS is based on the security features relative to other operating systems. It has to be considered how vulnerable the operating system is towards hackers and malware such as viruses, worms, trojan horses, spyware, and rootkits. As stated in various litterature found on the Internet( see [25], [34], and [32]) most of the viruses and other malware are made for Windows operating systems. This means that the Linux OS is a better alternative when taking into consideration, that the environment in which the web clients are to operate, have to be secure. In [32] it is stated that even though Linux is an open source software, it is not more susceptible against threats in comparison to Windows OSs. In fact, bugs in the Linux OS are corrected faster than in the case of Windows [32].
Furthermore, there is also the cost factor that points to the open source and cheap (if not free) Linux solution.
While Windows, among other because of its user friendly look, is most common as the Desktop operating system, the Linux OS is often used as the operating system of servers. But at the same time it is possible to get Linux GUIs such as GNOME and KDE to enhance its user friendliness for users that are used to the Windows environment. For the TOE in mind it is important to emphasise that the operating system on the TOE devices shall not prevent or make it more difficult for users to do their job.

In the following the security functions that are implemented in the web clients will be described.

**F.AUTH**

This security function addresses the authentication mechanisms in the TOE. Here, the authentication procedure of web clients is outlined. The authentication procedure consists of different mechanisms that each cover different aspects of the authentication. These aspects are as follows:

*a)* Identification of users

*b)* Identification of devices

Users, that are allowed access to the TOE via the web interface, can only do so from computers (stationary or laptops) that are configured by a TOE administrator. In order to allow only those computers to be used for accessing the TOE, the MAC addresses of the machines are used. The MAC address, which is a unique identifier of a network adapter, is used as *part of* the authentication mechanism of web clients, such that the web server provides services only for web clients with registered MAC addresses. If the network adapter or any other part of TOE devices are removed and placed in other devices (so a registered MAC address is used on another device which is not a device controlled by the TSF) in order to cause damage to the TOE, it will be considered as theft and vandalism and therefore seen as violation of the assumption A.PHYSICAL.

Identification of users is done by a username, a password, and an authentication token. The authentication token is a RSA SecurID Token [11] which generates an authentication code every sixty seconds. The generated code has to be concatenated to a personal password that has been given to the user by an administrator of the TOE.

**F.CRYPTOGRAPHY**

It is required that the web clients are equipped with a browser that supports the cryptographic functions and policies as stated in the ST. The browser chosen in this concrete application is "Mozilla Firefox 2" due to several factors:

- Secure features,
- open source, ie. free of cost, and
- good automated updates [9].

Beyond these factors, Firefox 2 is well suited for devices running a Linux OS environment.
But related to the cryptographic operations, most importantly Firefox 2 supports the SSL3.1/TLS1.0 cryptographic protocol for secure communication on the Internet. This means that any of the TLS cipher suites for cryptographic operation, stated in the ST are applicable. Thereby web clients can send authentication data SSL/TLS encrypted to the web server.

**F.SCAN**

As can be seen in figure 7.1, a part of the web client is constituted of a "Rootkit Scanner". A rootkit is a set of software tools that is used for concealing running processes, files or system data from the operating system. Thus rootkits are often used by attackers to hide malware like

backdoors and thereby gain and maintain access to systems [11]. Since the web client must be run on TOE devices that may not spread any spyware or do any malicious harm within the TOE, it is required that TOE devices run rootkit and vulnerability scans during initial start-up, periodically during normal operation, and at the request of an authorised user. This is configured by TOE administrators prior to handing out devices to TOE users. The rootkit scanner chkrootkit is used for this purpose.
Furthermore, scans for viruses and other malware is done by the EAL4+ evaluated Fortinet FortiGate[1] product[19] [8]. This product also provides the firewall functionality in the web clients.

**F.MANAGEMENT**
This security function in web clients is met by the components Red Hat Enterprise Linux WS and Firefox 2. The Red Hat OS provides functionalities for setup and configuration of applications and it manages the authorisation and authentication of users that use the computer (e.g. only administrators are allowed to configure and install programs).
The Firefox 2 contributes the management of cryptographic services.

Beside the costs related to the actual device that the web clients run on (price of the computer), the components that contribute largely to the costs of web clients are RSA tokens and Fortinet Fortigate. The functionality they provide are though required and therefore investing in these products is well spent money. Fortinet Fortigate is a very costly software but only 1 copy would be needed in order to install on all clients. There are though licenses that must be dealt with, but some enterprise discount is likely to make Fortinet affordable. This also goes for RSA Secure IDs. Tokens are relatively cheap, but of course it has to be taking into account how many users are suppose to use the system and how regular the tokens need replacement.

### 7.2.2 Web Server

The purpose of the web server is to provide the web clients with a web interface that enables users to monitor and control windmills securely. The web server thereby functions as intermediate link between web clients and windmills. The protocols for secure communication are HTTPS between the web clients and the web server, and SOAP between the web server and the windmills. The web server must comply with and be implemented according to these protocols.

[1] The Fortinet FortiGate software gathers several security protection tools into one program, in other words into an Unified Threat Management application. It contains for instance antivirus, intrusion detection, and firewall systems.

The operating system, running on the web server, has a central role in implementing the security functions in the TOE. Some of the security functions may be fulfilled directly by the OS or their implementation is influenced by the OS. The chosen operating system for the web server is Red Hat Enterprise Linux AS Version 3 Update 2, which is EAL3+ evaluated [19]. For the same reasons as for the web clients, this OS has been chosen (see section 7.2.1). Additionally, it must be pointed out that a Linux server is considered more reliable and stable in comparison with a Windows server [25] [32].
The Oracle Application Server 10g (EAL4 evaluated) runs the server-side of the DMC Application, which takes care of receiving and processing user requests and replies to/from windmills, as stated in the description of the ST TOE (see chapter 5). The DMC Application therefore interfaces with the Oracle DBMS which handles the data trail that the DMC Application interacts with.

The security functions that are present in the web server are described below. This includes specifying concrete components that have been found suitable to meet the security functions, and a description of how they satisfy the functions.

**F.BACK-UP**
: The security function of back-up is performed partly by the operating system and partly by the Oracle Database 10g software using the Oracle Secure Backup (OSB) [28]. Both TSF data (such as audit records) and user data are backed up. The back-up procedures are carried out as described in appendix B section B.7.1.

**F.ROLE**
: The operating system Red Hat Linux provides facilities for role based access control. But this is only for accessing the web server such that administrators can access the web server for management issues. The roles concerning access to the TOE through the web interface are provided for through the Oracle Application Server. Among other things, this component offers role based access control. The administrator can by logging on to the server OS, manage the roles of web interface users, ie. the services provided by the Oracle Application Server [19].

**F.AUDIT**
: The audit security function is taken care of by the audit functionality in the operating system. The Linux OS provides an audit capability that allows generation of audit records for the security critical events and provides tools for the administrative user to configure the audit subsystem and evaluate the audit records [17]. Audit records concerning the data trail is taken care of by the Oracle DBMS (also called Oracle Database 10g Enterprise Edition - EAL4 evaluated) which is a database management system that records every change in the TOE user data, ie. monitoring

and control data.

**F.AUTH**

Identification and authentication of administrators, that wish access to the web server, is taken care of by the operating system of the web server. Identification and authentication of users, that can be granted access to the services provided by the web interface for monitoring and controlling the windmills, is handled by the two-factor authentication mechanism of RSA SecurID Server together with the RSA SecurID token [11] and the Oracle Application Server.

Upon receiving authentication data from web clients, the Oracle Application Server matches the username, password, RSA token and MAC address of the device from which the request originates, before granting access to the TOE services. Username, passwords, and MAC addresses are stored as data files as part of the TSF data on the server. Whereas, the RSA token received is matched to that of the RSA SecurID Server.

Personal passwords are fixed and 8 characters long, and designed by administrators. After 3 unsuccessful logins, the user account is locked and can only be reopened by contacting an administrator.

**F.CRYPTOGRAPHY**

In order to set up the web server to provide secure communication for/to web clients, it is required that the web server provides SSL/TLS encryption for HTTPS communication (see figure 7.1). The setup for accepting HTTPS connections is done by creating a public key certificate for the web-server. The certificate is created by using the FIPS 140-2 validated OpenSSL tool. Using a FIPS 140-2 validated tool concerning cryptography is in compliance with the requirements for cryptographic operations as stated in the ST (see appendix B section B.6.1.2). The X.509 v3 standard for public key infrastructure (PKI) will be used as the standard for the certificate [33].

The TLS 1.0 shall be implemented by one of of the following cipher suites as stated in the ST:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA

The RSA public key encryption is implemented with 1024-bit key length. This is also applied to the SOAP communication between the web server and the windmills by secure SOAP over the HTTPS protocol.

Encryption of data in the database is provided by the Oracle DBMS [28]. Only the administrators are allowed to manage the cryptographic functions within the web server.

**F.SCAN**

In addition to the scanning mechanisms mentioned in the case of web

client devices (Fortinet Fortigate antivirus, intrusion detection, and firewall functionalities and rootkit scanner), the web server is scanned for changes of data caused by intrusion. The open source software Tripwire, which is a security and data integrity tool, is applied for monitoring and alerting on suspicious file change(s) [12].

**F.MANAGEMENT**
The operating system together with the Oracle Application Server, the Oracle DBMS, RSA SecurID Server, and the Fortinet FortiGate Firewal Antivirus components provide management facilities that can be used by administrators at any time. Administrators can access these through the OS by logging on. The management facilities cover the management functions stated in the FMT_SMF.1 SFR.

High security is required for the web server, and thus high costs are expected. This is also reflected in the components selected for the web server. Besides the server hardware, products such as the Oracle solution, RSA SecurID Server, Tripwire, SSL/TLS certificates and Fortinet FortiGate amount to a large expense but contribute sufficiently in meeting the security functionality requirements for the web server. Furthermore, regular expenses such as licences and updates of products must be accounted for.

### 7.2.3 Windmills

The windmills are equipped with a Programmable Logic Controller (PLC) microprocessor on which the windmill side DMC Application and the Windmill Driver run. The DMC Application interfaces with the Driver by exchanging data about windmill attributes (see chapter 5).
The PLC is favored among other alternatives because of its deployment in industrial use and because it is programmable [11]. PLCs are considered robust in terms of immunity to electrical noise and resistance to vibration and impact. It is well-known that PLCs are commonly deployed in windmills today precisely due the factors mentioned previously. These are highly relevant when considering windmills. The PLCs used in the Windmill DMC System support the Ethernet TCP/IP communication standard.
Implementing the windmill-side DMC application as a server program on the PLC may at first sight not be seen as the best option, because of server programs being relatively bigger than client programs. But in this case, the server program is not very big. It only maintains a connection to the web server (single thread) and be able to receive and send small SOAP objects (basically simple text (XML) strings). Therefore it is found suitable to implement this relatively

small server program on the PLC. The implementation of security functions in the windmills will be described in the following.

**F.AUTH**

In order for the web server and the windmills to verify who they communicate with, mutual authentication is needed. This is provided by the TLS1.0 protocol by implementing PKI for mutual authentication. The standard used for the certificates is the X.509 v3 standard for PKI. The authentication is taken care of by the DMC application on the web server and the cryptographic chip in the windmills (see below) by verification of each others certificates.

**F.CRYPTOGRAPHY**

The security function for cryptography is relevant in windmills since there is a data flow between the windmill and the web server. The SOAP service is deployed over HTTPS in order to obtain confidentiality and integrity of data. The TLS1.0 is implemented with one of the TLS_RSA_WITH_AES_128_CBC_SHA or the TLS_RSA_WITH_AES_256_CBC_SHA cipher suites. The RSA public key encryption is implemented with 1024-bit key length. The X.509 v3 standard for public key infrastructure (PKI) will be used as the standard for the certificates [33].

Since the PLC has limited processing capabilities and resources, and thus may not be able to handle the big computations that are required by the cryptographic operations, a cryptographic chip is installed in each windmill. Hereby, all above mentioned cryptographic operations are taken care of by this cryptographic chip. No concrete product is selected for this cryptographic chip due to time limitations for research and hereunder lack of knowledge of any evaluated product. A further description of how a cryptographic chip can be applied is sketched in [23].
An alternative approach would be to make the front-end handle the cryptographic operations. With this approach the connections within the local network in the windmill park are not encrypted. This is not in contrast with the ST, because the ST includes the A.PHYSICAL assumption such that it is assumed that no unauthorised entity is allowed entrance to the windmill park area and the wired connections thereby are secured. This approach requires that the front-end can handle maybe several SSL transactions simultaneously which requires a lot of computational resources and can therefore create a big burden for the front-end. In order to avoid bottlenecks cryptographic chip(s) could be installed in the front-end.

**F.MANAGEMENT**

This security function is relevant in windmills since it is necessary to manage the cryptographic functions (ie. management of the cryptographic

chip) that apply in transmission of data between the web server and the windmills, ie. the above mentioned security functions. It is only the administrators of the TOE that are allowed to manage these.

Costs related to windmills encompass mainly the cryptographic challenges since the PLCs on the windmills have limited processing capabilities. The solution shown in figure 7.1 requires purchasing and installment of cryptographic chips for each windmill. In the case of the alternative solution where the front-end takes care of the cryptography operations, the costs will substantially be lower. But it is preferred that data is encrypted all the way to the windmills because of the risk that an intruder can enter the windmill park area and fiddle with the connections in spite of the A.PHYSICAL assumption. Taking into consideration how much a windmill costs, acquiring a cryptographic chip in each windmill is "peanuts".
Additionally, costs related to the front-ends in each windmill park may also be taken into considerations. Though, it doesn't contribute with large expenses.

## 7.3 Conformance Claim

This example of an implementation representation for the Windmill DMC System is in conformance with the "Secure Windmill Distributed Monitoring and Control System ST" since the requirements stated in this ST are met in this implementation representation.

## 7.4 Design Conclusion and Comments

This chapter outlines a design proposal that claims conformance to the developed ST in appendix B. From this it can be concluded that the requirements stated in the ST are realistic and can be used for actual concrete design and implementation.

This implementation representation provides suggestions to how the identified security functions can be implemented in a concrete application of a Windmill DMC System.

The outlined design in this chapter should be seen as a first step in a complete design of the TOE. The design is informal and is just a rough sketch of components that constitute the TOE. Due to time limitations and the size of

the TOE this is seen as satisfactory.

The design and the components used for implementing the requirements stated for the TOE in the ST, is based on a standard web service solution, with Linux as operating system and Oracle as application service provider and database management system platform.

The concrete components that make up the proposed design were selected on basis of relevant security facilities that they offer, the cost, and their application. It has been tried to choose products that have been evaluated at least at EAL3 evaluation level in order to obtain a TOE which in its wholeness can be EAL3 evaluated. But yet, some of the selected products are evaluated at a lower level or not evaluated at all. It was necessary to include these in the design because the functionality they provide was needed and there was no other EAL3 evaluated equivalent product available. So, though not all products are EAL3 evaluated, the TOE can still be evaluated at EAL3, since it is the composition of the components that make the TOE EAL3 equivalent.

In the design example the costs and resources have been taken into account when selecting components when possible. It is realised that security does not come for free and this is also reflected in the suggested design where components may be selected because of their capabilities rather than how much they cost.

Chapter 8

# Conclusion and Comments

An outline for a CC approach in designing a secure system has been presented in this paper. The approach adopts the idea from traditional software engineering which is an iterative process of development that starts with an abstract specification, moves on to a concrete specification and ends up with an actual design. Within the context of the CC the abstract specification is equivalent to a PP, the concrete specification to a ST, and the design to an implementation representation.

The CC approach is applied for a windmill DMC system that starts with a rough sketch of a general DMC system consisting of web clients, several servers, and windmills. This architecture is used as the TOE for which the PP is developed.
For development of the ST, the TOE is constrained into consisting of web clients, windmills, and only one web server. This constraint was taken due to the scope and time limitations of this project.

The PP contains an analysis of the PP TOE in terms of a threat scenario and objectives from which security requirements are derived. The PP is defined relatively broad since it must be able to address any type of windmill DMC system.
When identifying requirements it was a general rule to select relevant and as few as possible components that could sufficiently cover the specified objectives.

There was no ultimate solution when selecting which requirements should be aimed for. A lot of requirements could be included, but some care must be taken in order not to set up unachievable requirements in real life for the system. This also has impact on how extensive the PP is. If the PP is too extensive no ST will be able to claim conformance to it.

The ST, which claims conformance to the PP, is built up similarly as the PP, except that the ST TOE is more detailed so threats, objectives, and requirements in the ST are more fine grained. Furthermore, the ST includes security functions that the TOE must implement in order to live up to the requirements. Even though the requirements in the ST have been constrained, developers have still been given the opportunity to develop several different implementation representations and concrete designs from the ST.
It has been up to the developers of the ST to make appropriate and realistic value assignments and selections for the security requirements.

A concrete example, in form of an implementation representation, of a windmill DMC system has been used to show that it is possible to design a secure system from the PP and ST when applying the CC approach. The example is based on a standard web service solution.
The implementation representation claims compliance to the ST, ie. meets the specified requirements and implements the security functions. From this example it can be concluded that the requirements stated in the ST are realistic and can be used for actual concrete design and implementation.

As assurance level EAL3 is found most suitable. This choice was made upon the context and environment in which the TOE is to operate and the security that is aimed for in these kind of systems.

Throughout the project development costs of design and implementation of such a system has been considered and been taken into account. It is realised that security does not come for free and this is also reflected in the development process of the PP, the ST, and the suggested implementation representation.

At the end of this project it can be concluded that the CC approach taken for designing secure systems is applicable. The approach is good because developers of a system are guided through different aspects of security for the system in mind. The system is analysed systematically and requirements to the system are described with a standardised language which is convenient for both developers and customers when they talk about security. This way misunderstandings are avoided and the security requirements are described exact.
Furthermore, a successful design of a windmill DMC system has been achieved, which further confirms that the CC approach applied in this project is a good approach for designing secure systems.

Appendix A

# Protection Profile (PP)

## A.1 PP Introduction

The stated PP covers windmill distributed monitoring and control systems (from now on denoted *WDMC*). The system is an IT system that makes it possible to monitor and operate windmills by gaining access to windmill data. This PP has been developed in order to identify and specify security and assurance requirements that are needed to protect such systems. The WDMC PP is a general protection profile made to suit an abstract design of such systems.

The primary audiences of this PP are: organisations that have something to do with windmills and wish to develop and deploy a WDMC system and/or organisations that wish to revise an already installed WDMC system for security requirements and update for any unforeseen new threats that may not be accounted for.

Naming conventions used in this PP:

**Assumptions**

TOE security usage assumptions are given names beginning with **"A."**, e.g. A.NO_EVIL.

**Threats**
TOE security threats are given names beginning with **"T."**, e.g. T.MODIFICATION.

**Policies**
TOE organisational security policies are given names beginning with **"P."**, e.g. P.TRAIN.

**Objectives**
Security objectives for the TOE and the TOE environment are given names beginning with respectively **"O."** and **"OE."**, e.g. O.DATA_INTEGRITY and OE.TRAIN.

### A.1.1 PP Identification

| | |
|---|---|
| Title: | Windmill Distributed Monitoring and Control System CC Protection Profile |
| Authors: | Vikas Vohra and Shekoufeh Khodaverdi |
| Publishing date: | 12th February 2007 |
| Version: | 1.0 |
| CC version: | This PP claims conformance to Common Criteria for Information Technology Security Evaluation (CC) Version 3.1, parts 2 and 3. |
| Evaluation Level: | Evaluation Assurance Level (EAL) 3 with no augmentation. |

### A.1.2 PP Organisation

The WDMC PP is organised as follows:

**Section A.1** gives an introduction to the WDMC PP, including relevant information for further reading and understanding of the WDMC PP, such as the Target of Evaluation (TOE) overview and abbreviations used in the PP.

**Section A.2** contains a description of the TOE and the environment it is to operate in.

**Section A.3** describes the security environment in which the TOE is to be deployed. This contains analysing potential threats, necessary security assumptions that must be present, and organisational security policies. In other words, this section identifies a threat scenario for the TOE.

**Section A.4** will contain identified security objectives. This includes objectives for both the TOE as well as the environment.

**Section A.5** identifies security functional and assurance requirements (SFRs and SARs) derived from the CC part 2 and 3, that must be enforced upon the TOE. Furthermore, the section identifies the requirements that are levied on the TOE environment.

**Section A.6** provides the rationale to illustrate that the security objectives for the TOE and its environment satisfy the identified threats, assumptions, and policies. Furthermore a rationale to show that the listed set of requirements are sufficient to meet each objective, and that each objective is covered by at least one requirement component, will be pointed out.

The content of this PP is in accordance with the guidelines stated in the CC part 1 appendix B [3]. This document is not an independent document and should be, where necessary, referred to the CC documentation for additional information and guidance. This is especially relevant in regards to the SFRs and SARs.

### A.1.3 TOE Overview

The TOE is capable of monitoring and controlling geographically distributed windmills and should be used for this purpose. The architecture of the TOE network is an open architecture where communication between the parts of the TOE is done through the Internet. This PP requires privacy and integrity of communications over the Internet using secure cryptographic algorithms. Furthermore, this PP addresses security requirements for a TOE that provides monitoring and controlling of windmills for users via a web-based interface. The security features of the TOE include identification and authentication, accountability and auditing, management, encryption, and data protection and integrity.
The assurance requirement specified in the PP are EAL3 compliant with no extra augmentation.

### A.1.4 CC Conformance Claims

This PP does not claim conformance to any other PP or requirements package since there are no validated PPs or packages related to the WDMC System.
The PP and the TOE claim conformance to CC part 2 and 3.
Any ST claiming conformance to this PP, must provide clear evidence that it meets requirements stated in this PP.

### A.1.5 Abbreviations

The following abbreviations are used throughout the WDMC System CC Protection Profile:

| | |
|---|---|
| **CC** | Common Criteria for Information Technology |
| **CRC** | Cyclic Redundancy Check |
| **DAC** | Discretionary Access Control |
| **DMC** | Distributed Monitoring and Control |
| **EAL** | Evaluation Assurance Level |
| **IT** | Information Technology |
| **MAC** | Mandatory Access Control |
| **OSP** | Organisational Security Policy |
| **PP** | Protection Profile |
| **SAR** | Security Assurance Requirement |
| **SFP** | Security Function Policy |
| **SFR** | Security Functional Requirement |
| **ST** | Security Target |
| **RBAC** | Role-Based Access Control |
| **TOE** | Target of Evaluation |
| **TSF** | TOE Security Functionality |

**TSFI** TSF Interface

**TSP** TOE Security Policy

**WDMC** Windmill DMC System

**WPP** WDMC System CC Protection Profile

## A.2 TOE Description

The TOE is a DMC windmill system which consists of windmills, software/hardware to control and monitor the windmills, and servers that handle user requests concerning monitoring and control of windmills.

### A.2.1 DMC Systems

The whole idea behind DMC systems is to monitor and control devices distributed geographically. DMC systems offer the advantage of centralised control, which is why their popularity has increased rapidly within recent time. Monitoring and controlling a lot of devices from a control centre is highly preferable rather than having to maintain several devices at a time. This also provides the opportunity to make devices operate synchronised in order to obtain better results. For instance windmills can be optimised in order to produce the desired amount of electricity by adjusting electricity production on different mills through regulation of various windmill attributes from one place.

Furthermore DMC systems also give an excellent opportunity for systematically keeping track of any changes in data. This is due to the fact that all data concerning requests about the status or control of devices is stored in a *data trail*. The data trail thereby provides an overview of the activity and the responsibles for the activities in the system. The amount of data that should be stored depends on the actual use of the system. This is also applicable for length of time and quality of data to be stored.
The data trail is the backbone of such DMC systems since it supplies definite evidence of activities in the system. If it is lost the system will definitely lose its value, become more vulnerable against threats, and in worst case break down.

So the purpose of any DMC system is to:

*a)* Monitor and control devices in the system in a distributed way;

*b)* Keep record of changes in the system, ie. ensuring proof/evidence through the data trail.

Often the transfer of data and commands in such DMC systems use some form of Ethernet or closed architecture in which a high level of security can be maintained. If instead an open architecture, like the Internet, is deployed for transmission it would make the system more vulnerable to attacks and threats. But on the other hand it would make it more versatile by also allowing access to the system outside the control centre, e.g. an inspector of devices out in the field [26] and any place with Internet connection.

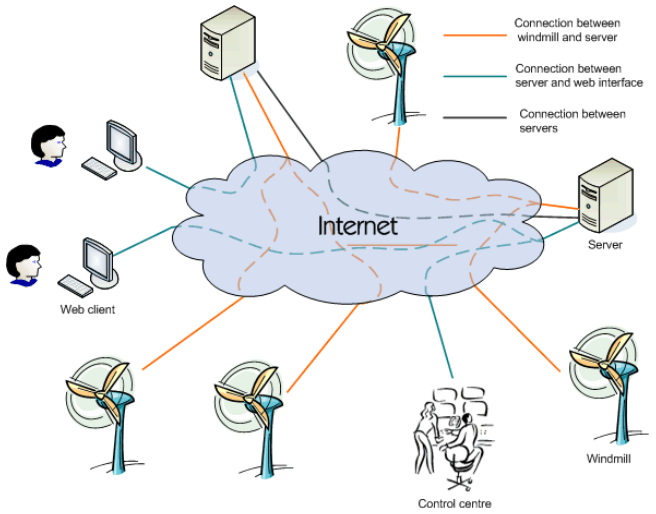


Figure A.1: The overall structure of the TOE system consisting of web interface, server, and windmills.

### A.2.2 The general windmill DMC System Model

The DMC model in this PP is built up such that the TOE can be accessed both from a central control centre and out in the field. This means the general DMC model will be accessed through an interface provided by a web service over the insecure Internet media.
In figure A.1 the distributed feature of the windmill control system can be seen. As shown on the figure users can access data about windmills via servers. The servers provide monitoring and control services through a web-interface.

Servers can also be interconnected in order to get data about windmills they do not have information about. The Internet is used as media to connect entities in the system.
The way operations are performed rely on which specific action (ie. monitor or control) is performed and by which user role (see section A.2.5). The operations the DMC system should satisfy are:

*a)* Monitoring and control devices in the system in a distributed way and

*b)* keeping record of changes in the system, ie. ensuring proof/evidence through the data trail.

The flow of data in the system when operations are performed is sketched in figure A.2. As shown, the data flows can be categorised into following:

- DMC input/output device - server
- Server - data trail
- Server - windmill

Users can through DMC devices generate a data flow to a server containing requests about monitoring or control of a windmill. The server processes the request by sending a request to the windmill in question. The windmill replies with a response which is again processed by the server and passed on to the output device of the user. Evt. the servers interact in order to get in connection with the windmill in question (this is also described previously in this section). Upon receiving requests and responses the server interacts with the data trail by reading/writing into the data trail.
Related to the operations of the DMC system, both requests and responses can be regarded as monitoring or control actions by users. While as the record into the data trail operation is the interaction between a server and the data trail.

### A.2.3 TOE Data

In general data can be grouped into user data and data that is related to the security functionality of the TOE. User data is made by and used for users of the TOE and includes monitoring and control data since these operations are performed by users. Whereas data related to the security functionality (such as authentication data and audit records) of the TOE is made and used by the

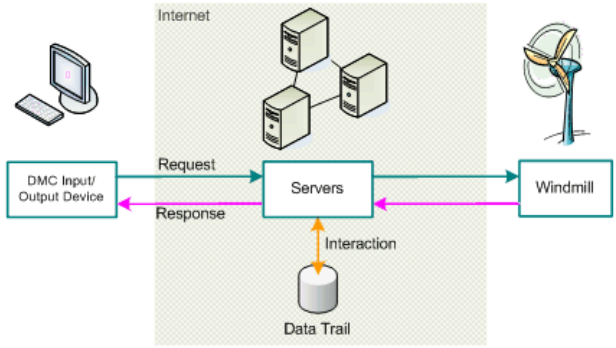


Figure A.2: Data flows in the general DMC model.

TSF (TOE security functionality) in order to ensure TOE security. The TSF is that part of the TOE which is in charge of all security within the TOE. Its goal is to implement all security functional requirements (SFRs). With this division of TOE data, the data that the data trail contains can be categorised into being partly user data (windmill monitoring and control data) and partly TSF data (audit records).

### A.2.4 TOE Devices

The flow of data in the general DMC system is caused by actions being executed on input/ouput devices. The following list of devices is an outline of potential DMC input/output devices:

- Computers and laptops
- PDAs
- Smartphones
- etc.

It is important to take into account which devices are used in the DMC system, since some devices may be at greater risk than others and thus jeopardise the security of the system differently.

### A.2.5 TOE Roles

Another factor to consider is the roles users of the DMC system are assigned. Each role interacts in its own individual way with the system and thereby uphold individual rights and permissions when security is concerned.
Following roles are identified:

**Service technician -** The role of a service technician is to make sure that apparatus and instruments of the devices work correctly, ie. responsible in maintaining the physical security and has nothing to do with the IT functionality. In order to judge whether there is any malfunction in the system, it has to be possible for the technician to read (monitor) the status of the devices in the DMC system. The workplace of a technician in the Windmill DMC System is at the windmills out in the field.

**Operations engineer -** The operations engineer's job is to monitor and control the DMC system. Depending on regional divisions, number of operations engineers, workload, etc. rights and privileges to access data of the DMC system can vary from engineer to engineer. In other words operations engineers may monitor and control a subset of the system. Operations engineers are furthermore responsible for validating that both monitoring data and control data in the DMC system are correct. The operations engineer can perform the tasks from either the control centre or out in the field.

**Administrator -** The administrator is the one who is responsible for the overall functionality and security of the DMC system. The administrator of the DMC system owns rights and privileges to perform changes (installation and configuration) in order to maintain the functions and security of the DMC system. In addition the administrator is responsible of user accounts, ie. creation of new user profiles with appropriate rights and privileges as well as maintenance of already existing user profiles.

The roles of the DMC system could be either static or dynamic, ie. the rights and privileges of a role can change over time or not. A dynamic model would be more suitable if frequent occurrence of promotion/degradation and thereby change in range of responsibilities among roles is present.
When possessing a role it is obviously clear that individuals that own that role are competent and trustworthy to carry out the work and responsibility that is demanded.

## A.3 TOE Security Environment

In this section assets, threat agents, threats, assumptions, and OSPs for the TOE security environment will be outlined.

### A.3.1 Assets

In regards to the purpose of the TOE:

*a)* Monitor and control devices in the system in a distributed way.

*b)* Keep record of changes in the system, ie. proof/evidence through the data trail.

the assets that need protection are

- the data trail,
- monitoring data,
- and control data.

Any malicious modification in these data jeopardises the security of the TOE. Below is listed an overall description of the possible critical points in the model, where the assets are most vulnerable:

**Servers -** The data trail, that resides on servers, is valuable to the TOE since it contains significant information in order to uphold the security of the TOE. Furthermore all data received and sent through the servers are also vulnerable. Servers (and thereby all the assets) are vulnerable to system breakdown, physical damage or attacks by malicious users or programs.

**Windmills -** Windmills are other potential weakness points in the TOE. It is expected that windmills send correct monitoring data when requested and upon receiving control data they act correspondingly. Within a windmill monitoring data may be read incorrectly due to malfunction of monitoring apparatus or data may be modified by unauthorised entities. Furthermore, received correct control data may be modified by malicious activity in the windmill.

**Input/output devices -** These devices are one of the critical points in the model because they might be infected with malicious programs that could harm the TOE when users gain access to the TOE through these devices.

**Connections -** Since the connections in the TOE are established over an insecure media, they are potential targets for attacks. Both monitoring data and control data are threatened by data loss, being read and/or modified.

### A.3.2 Threat Agents

The threat agents are divided into 2 groups: *internal attackers* and *external attackers*.

**Internal attackers** are entities within the company itself.

**External attackers** are correspondingly entities outside the company borders.

Hereinafter the term *attacker* will cover both groups of threat agents.
There could be several reasons for wanting to break into the TOE and gain access to valuable TOE data. Among these could be jealousy, competition, industrial espionage, revenge, or fun.
This leads to the observation that a threat agent can be characterised by the factors such as expertise, available resources, and motivation [14]. It is obvious that an entity that is involved of all three factors is of greater threat to the TOE than an entity with lack of one or more of the factors. Observations show that the strongest factor is motivation. An entity with high motivation and a given level of expertise and a set of resources is more likely to launch an attack compared to another entity that has lower motivation but the same expertise and resources [14].
Having said that, the factors expertise and resources do not have so much impact on whether an entity launches an attack or not, ie.:

$$\textit{low expertise + resources + high motivation} \approx \textit{high expertise + resources + high motivation}$$

and

$$\textit{less resources + expertise + high motivation} \approx \textit{more resources + expertise + high motivation.}$$

### A.3.3 Threats

Potential threats are identified and listed below.

**T.MASQUERADE** ◇ Unauthorised user or process pretends to be another entity in order to gain access to data or other TOE resources.

**T.UNAUTHORISED_ACCESS** ◇ Mischievous users or programs may gain unauthorised access to data which they are not allowed to according to the TOE security policy.

**T.MODIFICATION** ◇ Attackers may try to maliciously fiddle with protected data of the TOE.

**T.UNATTENDED_SESSION** ◇ An attacker may gain unauthorised access to an unattended session.

**T.ACCIDENTAL_USER_ERROR** ◇ Users may make accidental errors that could jeopardise the security of the TOE.

**T.DATA_TRANSMISSION** ◇ An attacker may alter the transmission and thereby the confidentiality and the integrity of the data in the TOE.

**T.CRYPTO_LEAK** ◇ Key data or other executable code associated with the cryptographic functionality, which intends to protect the data in the TOE system, may be viewed, modified or deleted by mischievous users or programs.

### A.3.4 Assumptions

The following descriptions identify the assumptions needed for the TOE to be securely operational.

**A.CORRECT_DEVELOPMENT** ◇ The development of the TOE, ie. design, implementation, and test, is assumed to be carried out correctly so it results in a TOE without flaws and errors that may lead to exploration by malicious users or programs.

**A.NO_EVIL** ◇ It is assumed that administrators have no evil intentions and that they are appropriately trained to carry out their job correctly.

**A.PHYSICAL** ◇ The physical security of TOE is assumed provided in order to avoid physical loss or damage of the TOE due to external factors like

fire, theft, natural catastrophes etc. Thus by this assumption the security of the data and the functionality of TOE are preserved.

### A.3.5 Organisational Security Policies (OSPs)

The OSPs are a set of rules, practices and procedures imposed by the organisation to address security needs. Following OSPs are identified:

**P.AUTHORISED_USERS** ◇ The TOE can only be accessed by authorised users.

**P.USER_PRIVILEGES** ◇ Users have different rights and privileges to access TOE data.

**P.ACCOUNTABILITY** ◇ Users that are authorised access to TOE data shall be held accountable for their actions within the TOE.

**P.CRYPTOGRAPHY** ◇ Data in the TOE has to be encrypted following some standard cryptographic algorithms.

**P.TRAIN** ◇ Authorised users of the TOE shall be trained appropriately in operating the TOE.

## A.4 Security Objectives

Following will contain an overview of the security objectives which aim at countering identified threats and/or comply with any OSPs and assumptions that were identified in section A.3. The rationale for these objectives can be found in section A.6.

### A.4.1 Security Objectives for the TOE

**O.UNIQUE_IA** ◇ The TSF shall ensure that unauthorised access to data in the TOE is not allowed. This shall be done by unique identification and authentication of entities trying to gain access to the TOE.

**O.DATA_INTEGRITY** ◇ Unauthorised modification, theft, or deletion of TOE data (user data and TSF data) shall be prevented.

**O.ACCOUNTABILITY_AND_AUDIT_RECORDS** ◇ The TSF shall provide individual accountability for audited events. The audit records shall record date and time of action and the identity of the entity responsible for the action.

**O.CRYPTO_FUNCTIONS** ◇ The TSF shall implement approved cryptographic algorithms.

**O.ROLE_MANAGEMENT** ◇ The TSF shall provide a mechanism for administrators to control rights and privileges according to user roles.

**O.SESSION** ◇ The TSF shall provide mechanisms that lock sessions automatically when the activity in an open session has been idle in a predefined period of time. Furthermore it shall be possible for users to manually lock a session in order to avoid signing out. Users shall be able to unlock a session by re-authentication and just continue the session where it was left.

**O.BACK-UP** ◇ The TSF shall provide procedures for back-up of TOE data. The data trail must be recoverable at any time.

**O.VULNERABILITY_ANALYSIS** ◇ The TSF will undergo vulnerability analysis in order to verify that design, implementation and test of the TOE does not contain any flaws.

**O.CRYPTO_SECRECY** ◇ Key data or other executable code associated with the cryptographic functionality shall be kept secret.

**O.SELF_TEST** ◇ The TOE shall provide self-testing functionality for all TOE security functions which can detect security vulnerabilities in the form of flaws and intrusions.

### A.4.2 Security Objectives for the Environment

**OE.TRAIN** ◇ Training of administrators, operational engineers, and service technicians will be provided by the overall responsible of the TOE.

**OE.ISOLATION** ◇ Those responsible for the TOE shall provide isolation of physical parts of the TOE such that they are protected from physical damages, intrusion, and theft.

## A.5 Security Requirements

In this section all requirements to the TOE will be stated. This includes functional, assurance and TOE environment requirements. The corresponding rationale for these requirements can be found in section A.6.

### A.5.1 TOE Security Functional Requirements

This section provides functional requirements that must be satisfied by PP-compliant TOE. These requirements consist of functional components from CC part 2. The components have been identified to fulfill the security objectives stated in previous section. The rationale behind identifying these components can be found in section A.6.2. Notice that selection and assignment identifications of some components are left to the ST authors. Table A.1 gives an overview of selected SFRs.

| SFR | Description |
|---|---|
| FAU_ARP.1 | Security alarms |
| FAU_GEN.1 | Audit data generation |
| FAU_GEN.2 | User identity association |
| FAU_SAA.1 | Potential violation analysis |
| FAU_SAR.1 | Audit review |
| FAU_SAR.2 | Restricted audit review |
| FAU_STG.1 | Protected audit trail storage |
| FCS_CKM.1 | Cryptographic key generation |
| FCS_CKM.2 | Cryptographic key distribution |
| FCS_CKM.4 | Cryptographic key destruction |
| FCS_COP.1 | Cryptographic operation |
| FDP_IFC.1 | Subset information flow control |
| FDP_IFF.1 | Simple security attributes |
| FDP_ITT.1 | Basic internal transfer protection |
| FDP_SDI.1 | Stored data integrity monitoring |
| FIA_UAU.2 | User authentication before any action |
| FIA_UAU.3 | Unforgeable authentication |
| FIA_UAU.6 | Re-authenticating |
| FIA_UID.2 | User identification before any action |
| FMT_MOF.1 | Management of security functions behaviour |
| FMT_MSA.1 | Management of security attributes |
| FMT_MSA.2 | Secure security attributes |
| FMT_MSA.3 | Static attribute initialisation |
| FMT_MTD.1 | Management of TSF data |
| FMT_SMF.1 | Specification of Management Functions |
| FMT_SMR.1 | Security roles |
| FPT_AMT.1 | Abstract machine testing |
| FPT_ITT.1 | Basic internal TSF data transfer protection |
| FPT_RCV.2 | Automated recovery |
| FPT_STM.1 | Reliable time stamps |
| FPT_TST.1 | TSF testing |
| FTA_SSL.1 | TSF-initiated session locking |
| FTA_SSL.2 | User-initiated locking |

Table A.1: Overview of identified SFRs.

#### A.5.1.1 Class FAU: Security audit

**FAU_ARP.1** **Security alarms**

FAU_ARP.1.1 The TSF shall take [assignment: *list of actions*] upon detection of a potential security violation.]

**FAU_GEN.1** **Audit data generation**

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

- Start-up and shutdown of the audit functions;
- All auditable events for the [selection, choose one of: *minimum, basic, detailed, not specified*] level of audit; and
- [assignment: *other specifically defined auditable events*].

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

- Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
- For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: *other audit relevant information*].

**FAU_GEN.2** **User identity association**

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

**FAU_SAA.1** **Potential violation analysis**

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

- Accumulation or combination of [assignment: *subset of defined auditable events*] known to indicate a potential security violation;
- [assignment: *any other rules*].

**FAU_SAR.1** **Audit review**

FAU_SAR.1.1 The TSF shall provide [assignment: *authorised users*] with the capability to read [assignment: *list of audit information*] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

**FAU_SAR.2** **Restricted audit review**

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

**FAU_STG.1** **Protected audit trail storage**

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to [selection, choose one of: *prevent, detect*] unauthorised modifications to the stored audit records in the audit trail.

### A.5.1.2 Class FCS: Cryptographic support

**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: *cryptographic key generation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

**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: *cryptographic key distribution method*] that meets the following: [assignment: *list of standards*].

**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: *cryptographic key destruction method*] that meets the following: [assignment: *list of standards*].

**FCS_COP.1 Cryptographic operation**

FCS_COP.1.1 The TSF shall perform [assignment: *list of cryptographic operations*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

### A.5.1.3 Class FDP: User data protection

**FDP_IFC.1** **Subset information flow control**

FDP_IFC.1.1 The TSF shall enforce the [assignment: *information flow control SFP*] on [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*].

**FDP_IFF.1** **Simple security attributes**

FDP_IFF.1.1 The TSF shall enforce the [assignment: *information flow control SFP*] based on the following types of subject and information security attributes: [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*].

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*].

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

FDP_IFF.1.4 The TSF shall provide the following [assignment: *list of additional SFP capabilities*].

FDP_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly authorise information flows*].

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly deny information flows*].

**FDP_ITT.1** **Basic internal transfer protection**

FDP_ITT.1.1 The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to prevent the [selection: *disclosure, modification, loss of use*]

of user data when it is transmitted between physically-separated parts of the TOE.

**FDP_SDI.1** **Stored data integrity monitoring**

FDP_SDI.1.1 The TSF shall monitor user data stored in containers controlled by the TSF for [assignment: *integrity errors*] on all objects, based on the following attributes: [assignment: *user data attributes*].

#### A.5.1.4 Class FIA: Identification and authentication

**FIA_UID.2** **User identification before any action**

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

**FIA_UAU.2** **User authentication before any action**

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

**FIA_UAU.3** **Unforgeable authentication**

FIA_UAU.3.1 The TSF shall [selection: *detect, prevent*] use of authentication data that has been forged by any user of the TSF.

FIA_UAU.3.2 The TSF shall [selection: *detect, prevent*] use of authentication data that has been copied from any other user of the TSF.

**FIA_UAU.6** **Re-authenticating**

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions [assignment: *list of conditions under which re-authentication is required*].

### A.5.1.5 Class FMT: Security management

**FMT_MOF.1** **Management of security functions behaviour**

FMT_MOF.1.1 The TSF shall restrict the ability to [selection: *determine the behaviour of, disable, enable, modify the behaviour of*] the functions[1] :

a) *Functions implementing creation and recovery of back-ups;*

b) *Functions implementing role management, including administration and maintenance of delegated roles;*

c) *Functions implementing routines for identifying events that have to be audited and administration and maintenance of audit records;*

d) *Functions implementing methods for identification and authentication of users;*

e) *Functions implementing and maintaining access control methods;*

f) *Functions implementing methods for locking sessions;*

g) *Functions implementing secure procedures for data transfer;*

h) *Functions implementing procedures for ensuring physical security and its maintenance;*

i) *Functions implementing methods for self test and analysis of results from the testing;*

j) *Functions implementing timers and clock synchronisation;*

k) *Functions for managing any cryptography related issues; and*

l) [Assignment: *additional manageable functions*]

to *the administrators*[2].

**Application note:** *It is left to the ST author to assign additional manageable functions if needed and which operations to restrict.*

[1] [assignment: *list of functions*
[2] [assignment: *the authorised identified roles*

**FMT_MSA.1** **Management of security attributes**

FMT_MSA.1.1 The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] to restrict the ability to [selection: *change_default, query, modify, delete*, [assignment: *other operations*]] the security attributes [assignment: *list of security attributes*] to [assignment: *the authorised identified roles*].

**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: *access control SFP, information flow control SFP*] to provide [selection, choose one of: *restrictive, permissive,* [assignment: *other property*]] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [assignment: *the authorised identified roles*] to specify alternative initial values to override the default values when an object or information is created.

**FMT_MTD.1** **Management of TSF data**

FMT_MTD.1.1 The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear,* [assignment: *other operations*]] the[3]:

a) *Data trail;*

b) *Identification and authentication data;*

c) *Cryptographic algorithms and keys;*

d) *Audit records; and*

e) [assignment: *additional TSF data*]

to *the administrators*[4].

[3] [assignment: *list of TSF data*]
[4] [assignment: *the authorised identified roles*]

**Application note:** *It is left to the ST author to assign additional TSF data which needs management restriction and which operations to restrict.*

**FMT_SMF.1** **Specification of Management Functions**

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions[5] :

a) *Functions to create and recover back-ups;*
b) *Functions to administrate and maintain delegated roles;*
c) *Functions to maintain audit records and to identify events that have to be audited and administrated;*
d) *Functions to identify and authenticate users;*
e) *Functions to protect data by using access control methods;*
f) *Functions setting up and maintaining session locking attributes;*
g) *Functions that provide secure procedures for data transfer;*
h) *Functions ensuring physical security and its maintenance;*
i) *Functions for self testing and analysing results from the testing;*
j) *Functions to synchronise timers and clock;*
k) *Functions that manage cryptography related issues; and*
l) [Assignment: *additional manageable functions*].

**FMT_SMR.1** **Security roles**

FMT_SMR.1.1 The TSF shall maintain the roles[6]:

a) *Service technician;*
b) *Operations engineer; and*
c) *administrator.*

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

[5] [assignment: *list of management functions to be provided by the TSF*]
[6] [assignment: *the authorised identified roles*]

### A.5.1.6 Class FPT: Protection of the TSF

**FPT_AMT.1 Abstract machine testing**

FPT_AMT.1.1 The TSF shall run a suite of tests [selection: *during initial start-up, periodically during normal operation, at the request of an authorised user,* [assignment: *other conditions*]] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

**FPT_ITT.1 Basic internal TSF data transfer protection**

FPT_ITT.1.1 The TSF shall protect TSF data from [selection: *disclosure, modification*] when it is transmitted between separate parts of the TOE.

**FPT_STM.1 Reliable time stamps**

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

**FPT_TST.1 TSF testing**

FPT_TST.1.1 The TSF shall run a suite of self tests [selection: *during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions*[assignment: *conditions under which self test should occur*]] to demonstrate the correct operation of [selection: [assignment: *parts of TSF*], *the TSF*].

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of [selection: [assignment: *parts of TSF*], *TSF data*].

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

**FPT_RCV.2 Automated recovery**

FPT_RCV.2.1 When automated recovery from [assignment: *list of failures/service discontinuities*] is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.2.2 For [assignment: *list of failures/service discontinuities*], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

#### A.5.1.7 Class FTA: TOE access

**FTA_SSL.1 TSF-initiated session locking**

FTA_SSL.1.1 The TSF shall lock an interactive session after [assignment: *time interval of user inactivity*] by:

- clearing or overwriting display devices, making the current contents unreadable;
- disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 The TSF shall require the following events to occur prior to unlocking the session: [assignment: *events to occur*].

### A.5.2 TOE Security Assurance Requirements

The assurance level of this TOE is EAL3 compliant. Following will list the TOE security assurance requirements. The EAL consists of assurance components that each meet an assurance requirement. The components are taken from the CC part 3.

| *Assurance Class* | *Assurance components/SARs* | |
|---|---|---|
| ADV:Development | ADV_ARC.1 | Security architecture description |
| | ADV_FSP.3 | Functional specification with complete summary |
| | ADV_TDS.2 | Architectural design |
| AGD:Guidance documents | AGD_OPE.1 | Operational user guidance |
| | AGD_PRE.1 | Preparative procedures |
| ALC:Life-cycle support | ALC_CMC.3 | Authorisation controls |
| | ALC_CMS.3 | Implementation representation CM coverage |
| | ALC_DEL.1 | Delivery procedures |
| | ALC_DVS.1 | Identification of security measures |
| | ALC_LCD.1 | Developer defined life-cycle model |
| ASE:Security Target evaluation | ASE_CCL.1 | Conformance claims |
| | ASE_ECD.1 | Extended components definition |
| | ASE_INT.1 | ST introduction |
| | ASE_OBJ.2 | Security objectives |
| | ASE_REQ.2 | Derived security requirements |
| | ASE_SPD.1 | Security problem definition |
| | ASE_TSS.1 | TOE summary specification |
| ATE:Tests | ATE_COV.2 | Analysis of coverage |
| | ATE_DPT.1 | Testing: basic design |
| | ATE_FUN.1 | Functional testing |
| | ATE_IND.2 | Independent testing - sample |
| AVA:Vulnerability assessment | AVA_VAN.2 | Vulnerability analysis |

Table A.2: Security assurance components/requirements in EAL3[7].

#### A.5.2.1 Class ADV: Development

**ADV_ARC.1 Security architecture description**

**Developer action elements:**

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

**Content and presentation elements:**

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialisation process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

**Evaluator action elements:**

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

**ADV_FSP.3 Functional specification with complete summary**

**Developer action elements:**

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

ADV_FSP.3.2D The developer shall provide a tracing from the functional specification to the SFRs.

**Content and presentation elements:**

ADV_FSP.3.1C The functional specification shall completely represent the TSF.

ADV_FSP.3.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.3.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.3.4C For SFR-enforcing TSFIs, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.3.5C For SFR-enforcing TSFIs, the functional specification shall describe direct error messages resulting from security enforcing effects and exceptions associated with invocation of the TSFI.

ADV_FSP.3.6C The functional specification shall summarise the non-SFR-enforcing actions associated with each TSFI.

ADV_FSP.3.7C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

**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 SFRs.

### ADV_TDS.2 Architectural design

**Developer action elements:**

ADV_TDS.2.1D The developer shall provide the design of the TOE.

ADV_TDS.2.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

**Content and presentation elements:**

ADV_TDS.2.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.2.2C The design shall identify all subsystems of the TSF.

ADV_TDS.2.3C The design shall describe the behaviour of each SFR non-interfering subsystem of the TSF in detail sufficient to determine that it is SFR noninterfering.

ADV_TDS.2.4C The design shall describe the SFR-enforcing behaviour of the SFR-enforcing subsystems.

ADV_TDS.2.5C The design shall summarise the non-SFR-enforcing behaviour of the SFR-enforcing subsystems.

ADV_TDS.2.6C The design shall summarise the behaviour of the SFR-supporting subsystems.

ADV_TDS.2.7C The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.2.8C The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.

**Evaluator action elements:**

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

ADV_TDS.2.2E The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

#### A.5.2.2 Class AGD: Guidance documents

**AGD_OPE.1 Operational user guidance**

**Developer action elements:**

AGD_OPE.1.1D The developer shall provide operational user guidance.

**Content and presentation elements:**

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance 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.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

**Evaluator action elements:**

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

### AGD_PRE.1 Preparative procedures

**Developer action elements:**

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

**Content and presentation elements:**

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

**Evaluator action elements:**

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

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

#### A.5.2.3 Class ALC: Life-cycle support

**ALC_CMC.3 Authorisation controls**

**Developer action elements:**

ALC_CMC.3.1D The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.3.2D The developer shall provide the CM documentation.

ALC_CMC.3.3D The developer shall use a CM system.

**Content and presentation elements:**

ALC_CMC.3.1C The TOE shall be labelled with its unique reference.

ALC_CMC.3.2C The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.3.3C The CM system shall uniquely identify all configuration items.

ALC_CMC.3.4C The CM system shall provide measures such that only authorised changes are made to the configuration items.

ALC_CMC.3.5C The CM documentation shall include a CM plan.

ALC_CMC.3.6C The CM plan shall describe how the CM system is used for the development of the TOE.

ALC_CMC.3.7C The evidence shall demonstrate that all configuration items are being maintained under the CM system.

ALC_CMC.3.8C The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

**Evaluator action elements:**

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

### ALC_CMS.3 Implementation representation CM coverage

**Developer action elements:**

ALC_CMS.3.1D The developer shall provide a configuration list for the TOE.

**Content and presentation elements:**

ALC_CMS.3.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; and the implementation representation.

ALC_CMS.3.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.3.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

**Evaluator action elements:**

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

### ALC_DEL.1 Delivery procedures

**Developer action elements:**

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

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

**Content and presentation elements:**

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

**Evaluator action elements:**

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

### ALC_DVS.1 Identification of security measures

**Developer action elements:**

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

**Content and presentation 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.

**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.

### ALC_LCD.1 Developer defined life-cycle model

**Developer action elements:**

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

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

**Content and presentation elements:**

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

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

**Evaluator action elements:**

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

#### A.5.2.4 Class ASE: Security Target evaluation

**ASE_CCL.1 Conformance claims**

**Developer action elements:**

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

**Content and presentation elements:**

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

**Evaluator action elements:**

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

### ASE_ECD.1 Extended components definition

**Developer action elements:**

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

**Content and presentation elements:**

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

**Evaluator action elements:**

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

ASE_ECD.1.2E The evaluator shall confirm that no extended component can be clearly expressed using existing components.

**ASE_INT.1 ST introduction**

**Developer action elements:**

ASE_INT.1.1D The developer shall provide an ST introduction.

**Content and presentation elements:**

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall identify the TOE.

ASE_INT.1.4C The TOE overview shall summarise the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

**Evaluator action elements:**

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

ASE_INT.1.2E The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

**ASE_OBJ.2 Security objectives**

**Developer action elements:**

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

**Content and presentation elements:**

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

**Evaluator action elements:**

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

**ASE_REQ.2 Derived security requirements**

**Developer action elements:**

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

**Content and presentation elements:**

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

**Evaluator action elements:**

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

**ASE_SPD.1 Security problem definition**

**Developer action elements:**

ASE_SPD.1.1D The developer shall provide a security problem definition.

**Content and presentation elements:**

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

**Evaluator action elements:**

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

**ASE_TSS.1 TOE summary specification**

**Developer action elements:**

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

**Content and presentation elements:**

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

**Evaluator action elements:**

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

ASE_TSS.1.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

#### A.5.2.5 Class ATE: Tests

**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 elements:**

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

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

**Evaluator action elements:**

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

**ATE_DPT.1 Testing: basic design**

**Developer action elements:**

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

**Content and presentation elements:**

ATE_DPT.1.1C The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems in the TOE design.

ATE_DPT.1.2C The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

**Evaluator action elements:**

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

**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 elements:**

ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

**Evaluator action elements:**

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

**ATE_IND.2 Independent testing - sample**

**Developer action elements:**

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

**Content and presentation elements:**

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

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

**Evaluator action elements:**

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

ATE_IND.2.2E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3E The evaluator shall test a subset of the TSF interfaces to confirm that the TSF operates as specified.

#### A.5.2.6 Class AVA: Vulnerability assessment

**AVA_VAN.2 Vulnerability analysis**

**Developer action elements:**

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

**Content and presentation elements:**

AVA_VAN.2.1C The TOE shall be suitable for testing.

**Evaluator action elements:**

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

AVA_VAN.2.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3E The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

### A.5.3 TOE Environment Requirements

In this section the requirements for the TOE environment will be stated including the thoughts and rationale behind them.

**OE.TRAIN**

In order even to be considered for operating the TOE, a user must have assigned a role, identified in section A.2.5. This means that there must be some procedure for evaluation if a user can be assigned the role. These procedures should in principal be present in the assurance requirement called AGD_OPE.1 - Operational user guidance. It should specifically be stated how training shall be carried out in terms of which qualification tests users should go through, which certificates they should acquire, and which courses they should attend before they are assigned a role.

**OE.ISOLATION**

It was previously assumed (A.PHYSICAL) that physical protection was provided as a general assumption. Reasonable precautions like safety rooms, barriers and fences to sensitive parts of the TOE (see section A.3.3) are expected to be in order.
Although physical protection of the TSF is addressed in the "TSF physical protection (FPT_PHP)" family, it does not state any measures for encountering the actual damage or theft of parts. The components in the family only detect when/if something physical to parts happen and specify what should be done. This is not sufficient to fulfil the objective. But together with the stated general assumption the OE is satisfied to satisfactory extent.

It is important to keep in mind that the specified requirements for the operational environment of the TOE are guidance lines to reach a reasonable level of security in relation to cost and other resources available. Furthermore, it must be noticed that it is not possible to measure or test whether or how much the requirements for the environment are followed.

## A.6 Rationale

This section of the PP provides the rationale for the security objectives, security functional requirements (SFRs), and security assurance requirements (SARs).

### A.6.1 Security Objective Rationale

The following illustrates the mapping of security objectives to identified threats, assumptions, and policies. This mapping is sketched in table A.3. The table also illustrates that all identified threats, assumptions, and policies are covered by at least one security objective.

| | O.UNIQUE_IA | O.DATA_INTEGRITY | O.ACCOUNTABILITY_AND_AUDIT | O.CRYPTO_FUNCTIONS | O.ROLE_MANAGEMENT | O.SESSION | O.BACK-UP | O.VULNERABILITY_ANALYSIS | O.CRYPTO_SECRECY | O.SELF_TEST | OE.TRAIN | OE.ISOLATION |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T.MASQUERADE | × | × | × | | | | | | | | | |
| T.UNAUTHORISED_ACCESS | × | × | × | | × | | | | | | | |
| T.MODIFICATION | × | × | × | × | | | × | | | | | |
| T.UNATTENDED_SESSION | | | | | | × | | | | | × | |
| T.ACCIDENTAL_USER_ERROR | | | × | | | | × | | | | × | |
| T.DATA_TRANSMISSION | | × | | × | | | | | | | | |
| T.CRYPTO_LEAK | | | | | | | | | × | | | |
| A.CORRECT_DEVELOPMENT | | | | | | | | × | | | | |
| A.NO_EVIL | | | | | | | | | | | × | |
| A.PHYSICAL | | | | | | | × | | | | | × |
| P.AUTHORISED_USERS | × | × | × | | × | | | | | | | |
| P.USER_PRIVILEGES | | | | | × | | | | | | | |
| P.ACCOUNTABILITY | × | | × | | × | | | | | | | |
| P.CRYPTOGRAPHY | | | | × | | | | | × | | | |
| P.TRAIN | | | | | | | | | | | × | |

Table A.3: Mapping from security objectives to threats, assumptions, and policies.

#### A.6.1.1 Rationale to counter Threats

Following rationale demonstrates how the objectives counter the threats.

**T.MASQUERADE**

This threat is primarily countered by the O.UNIQUE_IA, O.DATA_INTEGRITY and O.ACCOUNTABILITY_AND_AUDIT objectives.

O.UNIQUE_IA provides means for unique identification and authentication of users before they are granted access to the TOE services. It is clear that by ensuring this you are aware of who has access to the TOE and know exactly that it is authorised entities that gain access and thereby prevent masquerade from occurring.

O.DATA_INTEGRITY covers another aspect of the threat. It states that unauthorised modification, theft, or deletion of TOE data (user data and TSF data) shall be prevented. The fact that no unknown entity can perform any malicious activity on data in the TOE contributes in preventing T.MASQUERADE. Data integrity of TOE data is vital in order to keep the TOE reliable and trustworthy.

O.ACCOUNTABILITY_AND_AUDIT is another objective contributing in meeting the T.MASQUERADE threat. By providing individual accountability for audited events it is possible to keep track of user actions in the TOE and appropriate precautions can be taken if suspicious activities are recorded.

O.SELF_TEST is meant as a last option in detecting a masquerade but is not a preventing measure. The objective can detect malicious activity so precautions can be taken in order to protect the TOE against future similar attacks and/or limit the damages.

**T.UNAUTHORISED_ACCESS**

This threat is countered by the O.UNIQUE_IA, O.DATA_INTEGRITY, O.ACCOUNTABILITY_AND_AUDIT, and O.ROLE_MANAGEMENT objectives.

Exactly same reasoning as described for T.MASQUERADE can be applied to show that the O.UNIQUE_IA, O.DATA_INTEGRITY, O.ACCOUNTABILITY_AND_AUDIT, and O.SELF_TEST objectives also counter this threat.

The O.ROLE_MANAGEMENT objective comes into play when roles have to be administrated. This includes rights and privileges. Which resources of the TOE that should be accessible to which roles in the TOE must constantly be revised. Furthermore, in a company there will constantly be new staff being employed and old being fired and areas of responsibility might also be changed at times. So management of roles is a necessity in order to meet the unauthorised access threat.

**T.MODIFICATION**

This threat is encountered by following objectives: O.UNIQUE_IA, O.DATA_INTEGRITY, O.ACCOUNTABILITY_AND_AUDIT, O.CRYPTO_FUNCTIONS, and O.BACK-UP.
If a malicious modification of data has occurred the O.UNIQUE_IA objective can contribute in identifying which entity has been causing this.

O.DATA_INTEGRITY covers another aspect of the threat concerned. It states that unauthorised modification, theft, or deletion of TOE data (user data and TSF data) shall be prevented. The fact that no unknown entity can perform any malicious activity on data in the TOE contributes in countering the threat.

O.ACCOUNTABILITY_AND_AUDIT provides means for individual accountability for audited events and thereby makes it possible to keep track of user actions in the TOE. Together with the O.UNIQUE_IA objective it enables detection of malicious modification of TOE data by going through the audit records.

The objective O.CRYPTO_FUNCTIONS provides cryptographic measures that preserve integrity and confidentiality of TOE data. Thus, it states that the TSF shall implement approved cryptographic algorithms. Encryption of data can prevent data from being viewed and modified.

Back-up of data is enforced by O.BACK-UP. This is found relevant to the modification threat because if any unauthorised modification should happen it is possible to recover previous states of data. Especially the data trail should be recoverable at any time.

Same reasoning as before can be applied for the O.SELF_TEST objective countering this threat.

**T.UNATTENDED_SESSION**

Unattended sessions are mainly covered by the objective O.SESSION. Besides this objective, the OE.TRAIN also has some influence on this threat.

O.SESSION provides demands for unattended idle sessions. The TSF

shall provide mechanisms for locking user sessions automatically after a given idle time. Furthermore, the objective makes it possible for users to manually lock sessions and return to the session through re-authentication.

The environment objective OE.TRAIN also plays a part in securing the TOE against this threat. Appropriate training of users of the TOE include making them aware of possible risks in leaving sessions unattended which can minimize the threat.

Same reasoning as before can be applied for the O.SELF_TEST objective countering this threat.

The OE.ISOLATION environment objective addresses the threat from another perspective. It counters the T.UNATTENDED_SESSION threat by physically isolate the devices on which the sessions are running. It is only possible to isolate the control centre and thereby prevent unwanted people to gain access to the control centre. So unattended sessions can be secured against in the control centre through isolation but this is not the case out in the field. It is not possible to isolate the whole world. This is why the isolation environment objective only partly counters the T.UNATTENDED_SESSION threat.

**T.ACCIDENTAL_USER_ERROR**

This threat can be met by the security objectives O.ACCOUNTABILITY-_AND_AUDIT, O.BACK-UP and OE.TRAIN.

The rationale behind the O.ACCOUNTABILITY_AND_AUDIT and O.BACK-UP objectives for meeting this threat is quite obvious. If any accidental user error is detected in the audit records or by users them selves, it is possible to investigate what has gone wrong through O.ACCOUNTABILITY_AND_AUDIT in audit records and eventually recover the TOE into a previous state, through the O.BACK-UP objective.

OE.TRAIN demands that users are aware of how to use the TOE correctly. Training procedures should make users competent and responsible when they obtain an access role in the TOE as stated in section A.2.5. Thus, the objective contributes in restricting the threat from causing damages to the TOE by minimizing accidental user errors through appropriate training.

The O.SELF_TEST objective partly covers the threat too by providing means for detecting an accidental user error testing and scanning the TOE.

**T.DATA_TRANSMISSION**

Transmission of data is a weakness point in the TOE. Therefore appropri-

ate measures have to be applied in order to protect the data in transfer. This is mainly obtained through data integrity and data confidentiality. Thus, the objectives O.DATA_INTEGRITY and O.CRYPTO_FUNCTIONS counter this threat.
O.DATA_INTEGRITY ensures that unauthorised modification, theft, or deletion of TOE data shall be prevented. This also goes in regards to the connections over which the data is sent. This could be done through CRC (cyclic redundancy check) data check. The fact that no unknown entity can perform malicious activity on data in the connections of the TOE contributes in preventing T.DATA_TRANS-MISSION.

O.CRYPTO_FUNCTIONS provides countermeasures against this threat by encrypting data such that the data can not be read or modified by unauthorised entities. It could be in form of SSL encryption or using digital signatures which are typical cryptographic functions. The objective as such aims at protecting data not only within the TOE itself but also data transmission connections and thus is related to this threat.

Same reasoning as before can be applied for the O.SELF_TEST objective countering this threat.

**T.CRYPTO_LEAK**
This threat is so specific that only one objective is identified to counter this threat. This is the O.CRYPTO_SECRECY security objective.
Objective O.CRYPTO_SECRECY ensures that any data related to the cryptographic functionality, such as keys, signatures, and algorithms is kept secret. Hereby, directives against loss, theft, or modification of data are provided.

### A.6.1.2 Rationale to Uphold Assumptions

Following rationale aims at demonstrating how the objectives uphold the assumptions.

**A.CORRECT_DEVELOPMENT**
The way to uphold this assumption is by enforcing vulnerability analysis and tests upon the TOE.

O.VULNERABILITY_ANALYSIS is included in order to verify that design, implementation, and test of the TOE do not contain flaws. These vulnerability analysis should be enforced upon the TOE on regular basis during the phases of design, implementation and test. Furthermore

O.VULNERABILITY_ANALYSIS ensures that the TOE has been analysed for vulnerabilities and that any vulnerabilities found have been removed or otherwise mitigated during the phases.

The objective O.SELF_TEST ensures that the TOE provides self-testing functionality for all TOE security functions (scans and tests) so that flaws and intrusions can be detected. This is done regularly when the TOE is operational.

**A.NO_EVIL**
The objective OE.TRAIN upholds this assumption. This means that administrators are well trained and competent through the OE.TRAIN objective. Since training can not be seen as part of the functionality of the TOE, and can not be directly evaluated, the objective is stated as an objective for the environment. This is a realistic choice as training procedures cannot be carried out by the TSF but rather the overall responsible for the TOE. Furthermore, in order to obtain a role it is implied that the individual is trustworthy.

**A.PHYSICAL**
Physical protection of the TOE is divided into 2 categories: *back-up of data* and *isolation of sensitive parts.*

The O.BACK-UP objective addresses the issue of data back-up. The objective states that the TSF will provide procedures for back-up of data of the TOE. The data trail especially must be recoverable at any time.

Protection and isolation of sensitive parts is addressed in the objective OE.ISOLATION. Notice again that the objective is stated as an objective for the environment. This is reasonable because of the fact that isolation cannot be done by the TSF but is something that the overall responsible for the TOE shall ensure. The objective states that isolation of physical parts of the TOE will be provided, such that the TOE is safe guarded against physical damages, intrusion, theft, etc.

#### A.6.1.3 Rationale to meet Policies

Following rationale demonstrates how the objectives meet the OSPs.

**P.AUTHORISED_USERS**
O.UNIQUE_IA meets this OSP since it ensures that only authorised entities are allowed access to the TOE.

The objective O.DATA_INTEGRITY ensures that unauthorised modification, theft, or deletion of TOE data shall be prevented. Therefore this objective contributes in meeting this policy.

O.ACCOUNTABILITY_AND_AUDIT also covers an aspect of the policy, since it takes into account that if any unauthorised access is detected in audit records, administrators will be notified so that they can fix the problem and cover up the security hole. If not a proactive approach, then at least it gives some contribution in meeting the policy.

Which rights and privileges users are assigned according to the role they obtain, must be maintained by management procedures concerning roles. Furthermore, new people get employed and some get fired, so constant updates must be managed. Thus, the objective O.ROLE_MANAGEMENT has an influence on the P.AUTHORISED_USERS policy.

**P.USER_PRIVILEGES**

User privilege policies are met by the objective O.ROLE_MANAGEMENT. It is the only objective that concerns the management of user roles and privileges. It ensures that the TSF can provide a mechanism to control rights and privileges according to user roles, as identified in section A.2.5.

**P.ACCOUNTABILITY**

The objectives that address this policy are: O.UNIQUE_IA, O.ACCOUNTABILITY_AND_AUDIT and O.ROLE_MANAGEMENT.

The O.UNIQUE_IA objective has an influence in ensuring this policy, since obtaining accountability requires that entities are identified in order to be able to relate them to their activity and actions done within the TOE.

Generation of audit records in the TOE is ensured by the O.ACCOUNTABILITY_AND_AUDIT objective. The audit events are associated with the identity of users and their rights and privileges. Therefore accountability and audit is closely related to the objective O.ROLE_MANAGEMENT. Thereby, both O.ACCOUNT-ABILITY_AND_AUDIT and O.ROLE_ MANAGEMENT have an impact on the policy.

**P.CRYPTOGRAPHY**

To meet this policy there are two objectives, O.CRYPTO_FUNCTIONS and O.CRYPTO_SECRECY, that each concern two different aspects of this policy.

The O.CRYPTO_FUNCTIONS addresses the implementation of the the cryptographic algorithms and functions. It enforces the TOE data to be encrypted following some standard cryptographic services. The services provide confidentiality and integrity of data in transit (connections) as well as in end points (e.g. windmills and servers).

O.CRYPTO_SECRECY covers a different aspect of the policy. The objective states that key data or other executable code associated with the cryptographic functionality shall be kept secret. This includes keys, signatures, algorithms, etc.

O.SELF_TEST provides a partial upkeep to the policy since a self test can, when everything else fails, detect a flaw/error in cryptographic related code and alarm some suspicious activity and appropriate measures can be taken in order to safe guard against threats.

**P.TRAIN**
The only objective addressing this policy is the environment objective OE.TRAIN. Training has nothing to do with the functional operation of the TOE, and the OE.TRAIN is therefore an objective that must be satisfied by the operational environment of the TOE. The objective ensures that authorised users that hold a role within the TOE have received proper training and thus are found competent and trustworthy to operate the TOE in a secure way. Furthermore, users will get continuous training when new functions are incorporated in the TOE.

## A.6.2 Security Requirements Rationale

This section states the rationale behind the IT security functional requirements and security assurance requirements.

### A.6.2.1 Security Functional Requirements Rationale

The rationale for why the security objectives are met by specified SFRs is explained below, and an overview is shown in table A.4. Furthermore, table A.4 reasons why stated requirements are sufficient in order to meet all objectives and that each objective is covered by at least one requirement.

**O.UNIQUE_IA**
The FIA_UAU.2, FIA_UAU.3, and FIA_UID.2 components ensure that a

user is identified and authenticated before being allowed any actions and thus identification and authentication of all entities are enforced by these components. FIA_UAU.6 ensures re-authentication when a user wants to resume a locked session.

The management of identification and authentication security functions and attributes are ensured by the FMT class (security management) from where the FMT_MOF.1, FMT_MSA.1, FMT_MTD.1, FMT_SMF.1, and FMT_SMR.1 components are selected.

The FMT_MOF.1 component specifies security functions and which users have the privileges to manage these [5].

Management of security attributes regarding identification and authentication in the SF is enforced by FMT_MSA.1.

FMT_MTD.1 takes care of management of TSF data, hereunder data concerning identification and authentication. This component restricts the ability to access the TSF data to only being available for authorised users that own special rights to read, modify, or delete TSF data.

Besides other management functions which the FMT_SMF.1 component specifies, the management function of identification and authentication is also included in this component.

**O.DATA_INTEGRITY**

Integrity of data in the TOE (user data and TSF data) is ensured by the FDP (user data protection) and FPT (protection of the TSF) classes.

With the FDP_IFC.1 component an information flow control SFP is specified. Additionally, this component defines the list of subjects (e.g. users, machines, or processes), information (e.g. email or network protocols), and a subset of the possible operations in the TOE that this policy shall be enforced upon.

FDP_IFF.1 enforces the information flow control SFP specified in FDP_IFC.1 based on types of subject and information security attributes. The list of subjects and information controlled under the indicated SFP, and for each of these their security attributes are specified in this SFR.

The component FDP_ITT.1 ensures protection of user data when it is transferred within the TOE, ie. via internal channels. Since the TOE in mind is a distributed system and is composed of various physically-separated parts (see section A.2.2) this component covers this kind of user

data protection.

In order to maintain the integrity of stored user data in the TOE, FDP_SDI.1 is selected. It ensures protection of stored data by monitoring user data stored in containers controlled by the TSF for specified integrity errors on all objects, based on defined user data attributes.

The underlying abstract machine is a virtual or physical machine upon which the TSF executes. In order to verify the security assumptions, such as memory capacity and correct mode of operation, made about the underlying abstract machine the FPT_AMT.1 component specifies the conditions under which the verification has to occur by the TSF.

FPT_ITT.1 ensures integrity of TSF data that is being transferred between physically-separated parts of the TOE via internal channels.

In order to meet this objective, management issues have to be considered too especially regarding security attributes. Therefore, the components FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_SMF.1, and FMT_SMR.1 are included in the list of SFRs enforcing the O.DATA_INTEGRITY objective.

**O.ACCOUNTABILITY_AND_AUDIT**

This objective is mainly enforced by the FAU (security audit) class. From this class the component FAU_ARP.1 is selected, because when violation of TOE security has been detected, the TSF has to take some actions in order to correct the security violation. Responsible persons shall be informed. This component is dependant on the inclusion of the FAU_SAA.1 SFR.

In order to have a security alarm functionality in the TSF it is necessary that the TSF knows what to react upon in case of violation of security. With the FAU_SSA.1 component a set of rules in monitoring the audited events is specified which then will be used by the TSF to indicate a potential violation.

The FAU_GEN.1 component of the security audit data generation (FAU_GEN) family defines requirements for the level of auditable events, and specifies the list of data that shall be recorded in each record [5].

FAU_GEN.2 is another component of the FAU_GEN family that ensures that the TSF is able to associate each auditable event with the identity of the entity that caused the event [5].
Since audit records are evidence of what has been going on in the TOE

and can be used to detect unwanted activity it shall be possible for administrators to read audit information from the audit records, thus the FAU_SAR.1 component is included.
In accordance with the O.ACCOUNTABILITY_AND_AUDIT objective responsible users shall have access to view audit records. No one is allowed to modify or delete audit records since the audit data is significant for the maintenance of the TSF security.

FAU_STG.1 ensures protection of audit records from unauthorised access, modification, and/or deletion.

Furthermore, in order to ensure reliable time stamps for auditing and security attribute expiration, the FPT_STM.1 is needed.

The same management issues as in previous mentioned objectives are taken care of by the FMT class for the same reasons.

**O.CRYPTO_FUNCTIONS and O.CRYPTO_SECRECY**
When ensuring these objectives, it is not possible to look at the objectives separately since they are related to each other, ie. a SFR ensuring one objective depends on the presence of another SFR that satisfies the other objective.
The CC provides the cryptographic support class (FCS) for ensuring cryptography measures in the TOE.

FCS_COP.1 addresses the O.CRYPTO_FUNCTIONS objective. Through this SFR all cryptographic operations are required to be performed in accordance with a specified algorithm and with a cryptographic key of specified size.
By specifying of the use of cryptographic keys, FCS_COP.1 depends highly on how keys are managed in the TOE. These dependencies (FCS_CKM.1 and FCS_CKM.4) are provided for when ensuring the O.CRYPTO_SECRECY objective.
Furthemore, FCS_COP.1 is also dependent of the presence of secure security attributes (FMT_MSA.2) because clearly cryptographic data such as keys are security attributes in them selves and thus only secure values are valid for these keys.

FCS_CKM.1 ensures that cryptographic keys are generated in accordance with a specified algorithm and key size. FCS_CKM.1 partly contributes to ensure objective O.CRYPTO_FUNCTIONS since it is not possible to ensure that cryptographic operations are performed properly if not cryptographic keys are appropriately generated.
Furthermore, FCS_CKM.1 also ensures O.CRYPTO_SECRECY since gen-

eration of keys must not be possible to be done by other parties. Any forgery of keys must not be possible and keys have to remain secret.

FCS_CKM.2 makes sure that cryptographic keys are distributed following a specified key distribution method. This is required since their exists several entities in the TOE model (ie. user/web clients, servers and windmills) that need to exchange data encrypted and mutually authenticate them selves to each other, and can only do so by exchanging cryptographic keys.

FCS_CKM.4 takes the destruction of cryptographic keys into account. It ensures that keys are destroyed appropriately with a clear destruction method. Just like generation, destruction contributes in ensuring both the objectives O.CRYPTO_FUNCTIONS and O.CRYPTO_SECRECY.

**O.ROLE_MANAGEMENT**

This objective is enforced by the security management class (FMT). The components included are FMT_MOF.1, FMT_MSA.1, FMT_MTD.1, FMT_SMF.1, and FMT_SMR.1.
The roles that are given to users have to be maintained by the TSF in order to keep the TSF secure. This is ensured by the FMT_SMR.1 component.

**O.SESSION**

Partly ensured by management requirements, the objective O.SESSION is also met by the SFRs listed below. The chosen components are needed in order to incorporate automatically and manually session locking. In order to manage security attributes of session locking some of the already described management SFRs are needed for the O.SESSION objective. These SFRs are the FMT_MOF.1, FMT_MSA.1, FMT_MTD.1, FMT_SMF.1, and FMT_SMR.1.

***FTA_SSL.1 - TSF-initiated session locking***

This component defines a session locking procedure carried out by the system. The period of time, that a session is allowed to be interactive before session locking, is also specified hereunder. The component specifies furthermore the events that should occur prior to unlocking of the session.

***FTA_SSL.2 - User-initiated locking***

This component defines a session locking procedure carried out by the user. The component specifies furthermore the events that should occur prior to unlocking of the session.

Both FTA_SSL.1 and FTA_SSL.2 have dependency to FIA_UAU.1 (Tim-

ing of authentication). Furthermore the FIA_UAU.6 (Re-authenticating) SFR is necessary in order to cover O.SESSION.

**O.BACK-UP**

Components found to fulfill the O.BACK-UP security objective are the FMT_SMF.1 and FPT_RCV.2 components.
Among several other management functions, FMT_SMF.1 provides the means for an administrator to ensure continued operation of the TOE, including back-up and recovery. Thereby it specifies management functions for creating and recovering back-ups.

In the TOE considered in this project it is the data trail that has to be recoverable at any time. Back-up of data trail should be done on a regular basis since loss of data could mean costly damages. Therefore an automated approach is to be preferred.

FPT_RCV.2 specifies a list of failures/service discontinuities that the TSF shall recognise and react upon automatically.
This SFR is dependant on the inclusion of the assurance requirement AGD_OPE.1 which provides operational user guidance for back-up procedures too.

**O.VULNERABILITY_ANALYSIS**

This objective is ensured for by the assurance requirement AVA_VAN.2. It is clear that this objective can not be enforced by any functional requirement (excluding management issues that are discussed in the above mentioned objectives). AVA_VAN.2 ensures that vulnerability analysis are carried out on a regular basis to ascertain the presence of potential vulnerabilities.

**O.SELF_TEST**

In order to preserve the security of the TOE, the TSF has to periodically test its functionality and analyse whether it still is secure or not. This includes both detection of unauthorised entities (e.g. worms, vira, and spyware) and detection of flaws and errors in the various parts of the TSF (e.g. servers, file systems, and sensors). The FPT (protection of the TSF) class ensures the O.SELF_TEST objective best since it focuses on protection of TSF data.

The FPT_TST.1 (TSF self test) component specifies conditions under which self test should occur and the integrity of which parts of the TSF should be verified. Thus the integrity and assurance of correct operation of TSF is preserved by this SFR. This component is dependant on FPT_AMT.1.

### A.6.2.2 Security Assurance Requirements Rationale

This section states the rationale behind the security assurance requirements. The security assurance components and requirements are derived directly from the EAL. The chosen EAL level is based on following reasoning:

Firstly, the two lowest assurance levels (EAL1 and EAL2) only reflect basic assurance. Secondly, to have a product at level 5 or higher (EAL5 - EAL7) it is needed to rely upon underlying systems, among other the operating system. This means that they also must have at least same assurance as the TOE itself. Since it is not within the scope of this project to analyse these underlying systems, these EALs are not considered for this TOE. This only leaves EAL3 or EAL4 to be considered.
When comparing the two levels, it is important to take into consideration what the purpose of the TOE is, and under which circumstances and environment it will be deployed (see section A.2).
Furthermore, when looking closer at the assurance components that are different at the two levels, it is noticed that stronger demands during development, especially tests and vulnerability analysis which are identified as profound security objectives and a vital part of the environment in which the TOE is deployed, are in greater focus at EAL4 (e.g. in the AVA_VAN family).
During development more assurance is given by EAL4 than EAL3 by requiring a design description, an implementation specification/representation, and improved mechanisms/procedures that provide confidence that the TOE will not be tampered with during development or delivery. Especially an outline for an implementation representation is exactly what is aimed for in this project.
Assessing the context in which the TOE is to operate further indicates the choice of EAL. The TOE is to operate within a rather closed environment by predefined known users/roles, ie. people with windmill knowledge and who are company authorised. Violation of security could have severe consequences financially and physically, and can affect the individual living being, because the windmills contribute electricity to the power system and their operation is important to the overall power supply[7]. So the TOE must ensure that windmills are functioning correctly. This means high assurance to its security is highly relevant.
Having said that, EAL4 would be the most likely choice, but there are some requirements defined in EAL4 which are beyond the scope of this project. This includes for instance considerations on how the TOE shall be delivered and demands for giving a subset of the actual implementation.
Therefore it is concluded that the level of assurance stated by EAL3 without any augmentation is found most appropriate and therefore chosen. Since a partly implementation representation of the TSF is aimed for in this project, the assurance requirement ADV_IMP.1 of EAL4 would have been ideal to include and

[7] Much like the power circuit breakdown stated in [22].

| | O.UNIQUE_IA | O.DATA_INTEGRITY | O.ACCOUNTABILITY_AND_AUDIT | O.CRYPTO_FUNCTIONS | O.ROLE_MANAGEMENT | O.SESSION | O.BACK-UP | O.VULNERABILITY_ANALYSIS | O.SELF_TEST | O.CRYPTO_SECRECY |
|---|---|---|---|---|---|---|---|---|---|---|
| FAU_ARP.1 | | | × | | | | | | | |
| FAU_GEN.1 | | | × | | | | | | | |
| FAU_GEN.2 | | | × | | | | | | | |
| FAU_SAA.1 | | | × | | | | | | | |
| FAU_SAR.1 | | | × | | | | | | | |
| FAU_SAR.2 | | | × | | | | | | | |
| FAU_STG.1 | | | × | | | | | | | |
| FCS_CKM.1 | | | | × | | | | | | × |
| FCS_CKM.2 | | | | × | | | | | | × |
| FCS_CKM.4 | | | | × | | | | | | × |
| FCS_COP.1 | | | | × | | | | | | × |
| FDP_IFC.1 | | × | | | | | | | | |
| FDP_IFF.1 | | × | | | | | | | | |
| FDP_ITT.1 | | × | | | | | | | | |
| FDP_SDI.1 | | × | | | | | | | | |
| FIA_UAU.2 | × | | | | | × | | | | |
| FIA_UAU.3 | × | | | | | | | | | |
| FIA_UAU.6 | × | | | | | × | | | | |
| FIA_UID.2 | × | | | | | | | | | |
| FMT_MOF.1 | × | × | × | × | × | × | × | × | × | × |
| FMT_MSA.1 | × | × | × | × | × | × | × | × | × | × |
| FMT_MSA.2 | | | | × | | | | | | × |
| FMT_MSA.3 | | × | | | | | | | | |
| FMT_MTD.1 | × | × | × | × | × | × | × | × | × | × |
| FMT_SMF.1 | × | × | × | × | × | × | × | × | × | × |
| FMT_SMR.1 | × | × | × | × | × | × | × | × | × | × |
| FPT_AMT.1 | | × | | | | | | | | |
| FPT_ITT.1 | | × | | | | | | | | |
| FPT_RCV.2 | | | | | | | × | | | |
| FPT_STM.1 | | | × | | | | | | | |
| FPT_TST.1 | | × | | | | | | | × | |
| FTA_SSL.1 | | | | | | × | | | | |
| FTA_SSL.2 | | | | | | × | | | | |

Table A.4: Rationale for requirements satisfying security objectives.

thereby augment EAL3 with this component. But because ADV_IMP.1 has dependency on other components of EAL4, this is abstained from.
The included assurance components are predefined by the CC, and listed in table A.2.
Notice that the assurance component AGD_OPE.1 covers all objectives. This is due to the given definition in [7] section 13.1 which states that this component is an operational user guidance document. It describes the security functionality provided by the TSF and gives instructions and guidelines, and helps to understand the TSF. Furthermore, it includes the security-critical information and actions required for its secure use.

Appendix B

# Security Target (ST)

## B.1 ST Introduction

The stated ST addresses secure windmill distributed monitoring and control systems, from now on denoted *WDMC*.
The system provides secure monitoring and control of windmills. This is achieved through a web server that provides means for users to control and monitor windmills securely over the Internet.
The WDMC ST is a specific security target made to suit a concrete design of such a system. This ST describes the TOE, the environment in which it is to operate, the threats against it, and the functionality required and provided to counter these threats.

### B.1.1 ST Identification

| | |
|---|---|
| Title: | Windmill Distributed Monitoring and Control System<br>CC Security Target |
| Authors: | Vikas Vohra and Shekoufeh Khodaverdi |
| Publishing date: | 12th February 2007 |
| Version: | 1.0 |
| CC version: | This ST claims conformance to Common Criteria for Information Technology Security Evaluation (CC) Version 3.1 and to the Windmill Distributed Monitoring and Control System CC Protection Profile |
| Evaluation Level: | Evaluation Assurance Level (EAL) 3 with no augmentation. |

### B.1.2 ST Organisation

The WDMC ST is organised as follows:

**Section B.1** gives an introduction to the WDMC ST. This includes an overview of Target of Evaluation (TOE), conformance claims, and abbreviations used.

**Section B.2** states a description of the TOE which this ST is intended for. The description will provide the reader with a general understanding of the architecture of the TOE and the environment which the TOE is dependent on.

**Section B.3** describes the security environment in which the TOE is to be deployed. This includes analysing potential threats, necessary security assumptions that must be present, and organisational security policies that must be present for the TOE. In other words, this section identifies a threat scenario for the TOE.

**Section B.4** will contain identified security objectives. This includes objectives for both the TOE as well as the environment.

**Section B.5** contains any components that have been extended.

**Section B.6** identifies security functional and assurance requirements (SFRs and SARs) that must be enforced upon the TOE. In the section any requirements that are levied on the TOE environment will also be identified.

**Section B.7** provides a TOE summary specification which purpose is to provide a description of how the TOE satisfies all the SFRs. It is also specified which general technical mechanisms that the TOE uses for this purpose in form of security functions and assurance measures.

**Section B.8** provides the rationale that illustrates that the security objectives for the TOE and its environment satisfy the identified threats, assumptions, and policies. Furthermore, a rationale to show that the listed set of requirements are sufficient to meet each objective, and that each objective is covered by at least one requirement component, will be pointed out.

### B.1.3 TOE Overview

The TOE is capable of monitoring and controlling geographically distributed windmills and should be used for this purpose. The architecture of the TOE network is an open architecture where communication between the parts of the TOE is done through the Internet. This ST requires confidentiality and integrity of data when communicating over the Internet. This is done using secure cryptographic algorithms.
Furthermore, this ST addresses security requirements for a TOE that provides monitoring and controlling of windmills for users via a web-based interface. The security features of the TOE include identification and authentication, accountability and auditing, management, encryption, and data protection and integrity.
The assurance requirements specified in the ST are EAL3 compliant with no augmentation.

### B.1.4 Conformance Claims

This ST claims conformance to the Common Criteria Information Technology Security Evaluation (CC) v. 3.1 part 2 and 3. Security functional requirements are based on components from part 2, while assurance requirements are based on an EAL (assurance package) which is constituted of components from part 3.
Furthermore, this ST claims strict conformance to the Windmill Distributed Monitoring and Control System CC Protection Profile by providing evidence

that requirements of the PP are met in this ST, ie. this ST is an instantiation of the PP.

### B.1.5 Abbreviations

The following abbreviations are used throughout the WDMC System CC Security Target:

| | |
|---|---|
| **AES** | Advanced Encryption Standard |
| **CBC** | Cipher Block Chaining |
| **CC** | Common Criteria for Information Technology |
| **CORBA** | Common Object Request Broker Architecture |
| **CRC** | Cyclic Redundancy Check |
| **DAC** | Discretionary Access Control |
| **DES** | Data Encryption Standard |
| **DMC** | Distributed Monitoring and Control |
| **EAL** | Evaluation Assurance Level |
| **FIPS** | Federal Information Processing Standard |
| **FTP** | File Transfer Protocol |
| **GIOP** | General Inter-ORB Protocol |
| **HTTP** | Hyper Text Transfer Protocol |
| **HTTPS** | HTTP using SSL |
| **ICE** | Internet Communications Engine |
| **IT** | Information Technology |
| **MAC** | Mandatory Access Control |
| **ORB** | Object Request Broker |
| **OSP** | Organisational Security Policy |
| **PKI** | Public Key Infrastructure |

| | |
|---|---|
| **PP** | Protection Profile |
| **RBAC** | Role-Based Access Control |
| **RC4** | Rivest Cipher 4 |
| **RSA** | Ron **R**ivest, Adi **S**hamir and Len **A**dleman Algorithm |
| **SAR** | Security Assurance Requirement |
| **SFP** | Security Function Policy |
| **SFR** | Security Functional Requirement |
| **SHA** | Secure Hash Algorithm |
| **SHS** | Secure Hash Standard |
| **SMTP** | Simple Mail Transfer Protocol |
| **SSL** | Secure Socket Layer Protocol |
| **ST** | Security Target |
| **TLS** | Transport Layer Security Protocol |
| **TOE** | Target of Evaluation |
| **TSF** | TOE Security Functionality |
| **TSFI** | TSF Interface |
| **TSP** | TOE Security Policy |
| **WDMC** | Windmill DMC System |
| **WPP** | WDMC System CC Protection Profile |

## B.2 TOE Description

The TOE is a DMC windmill system which consists of windmills, software/hardware to control and monitor the windmills, and a server that functions as intermediate link between user requests and the windmills.
The purpose of any DMC system is to:

*a)* Monitor and control devices in the system in a distributed way and

*b)* keep record of changes in the system, ie. assuring proof/evidence through the data trail.

The DMC model for the TOE is firstly built up in a way, so that the TOE can be accessed both from a central control centre and out in the field. This means the Windmill DMC System will be accessed through an interface over the insecure Internet media.
As illustrated in figure B.1, the architecture of the TOE model consists of web clients, only one web server which all entities in the TOE communicate via, and windmills. Even though only one server is included in the model, the distributed feature is still present since the entities in the TOE are physically distributed.

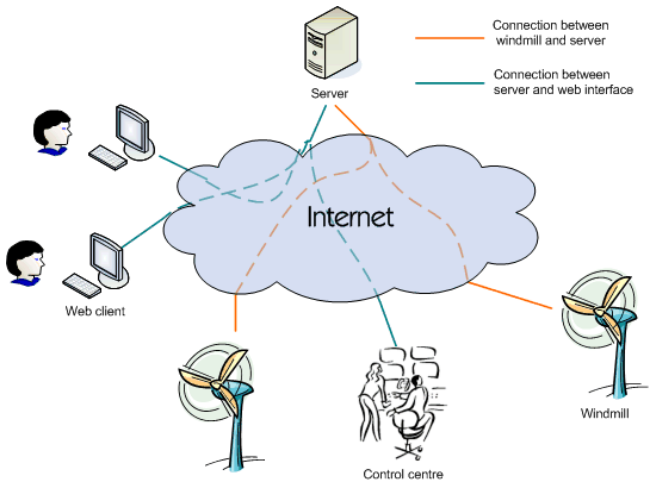


Figure B.1: The overall structure of the TOE consisting of web interface, a web server, and windmills.

Figure B.2 shows how the components of the TOE are related and gives an overview of the data flows in the TOE.
The communication between a web client and the web server is based upon the HTTP protocol (Hyper Text Transfer Protocol). For the communication between the web server and the windmills the SOAP[1] protocol is used.

[1]SOAP (Simple Object Access Protocol) is a simple XML-based protocol to let applications exchange information over HTTP.

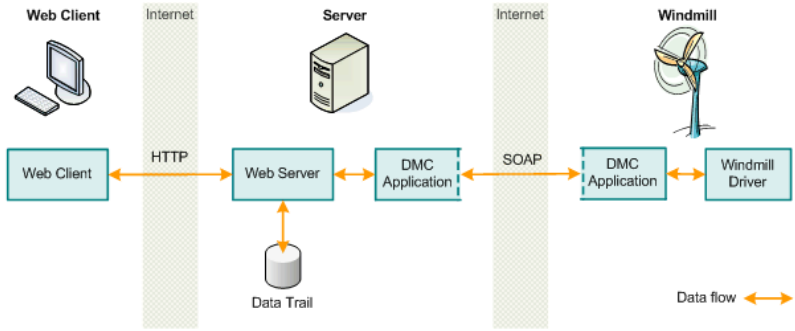


Figure B.2: A schematic overview of the TOE.

Below, the components/entities of the TOE are listed and described shortly:

**Web client**

The web interface is accessible through a web browser that supports HTTP communication. It provides a corporate login facility that is used to authenticate the user. The authentication is done by the web server before the user is allowed access to the TOE.
Upon successful authentication users are allowed to get an overview of the status of windmills. Furthermore, the web interface provides functions for changing windmill attributes.

**Server**

The server acts as a web server to the users that make use of the web interface to control and monitor windmills. Every time a user or another entity tries to gain access via the web interface the server registers the attempt into the data trail. The attempt is registered by noting when, from where, and by whom the attempt has been carried out.
As shown in figure B.2 users can after login send requests concerning monitoring and controlling windmills to the server, whereafter the server registers the requests into the data trail and subsequently passes on the requests to the DMC application which takes of the web server-windmill communication by implementing SOAP. The DMC application is divided into two: the server-side DMC application (running on the server) and the windmill-side DMC application (running on the windmill OS). The server-side of the DMC application transmits the requests to the windmill-side of the DMC application running on the windmills that the requests concern. The windmill-side of the DMC application then processes the requests by among other things making the Windmill Driver to act according to the commands of the requests.

After having processed the requests the DMC application sends back the replies. The web server registers the replies as responses to the corresponding received requests and then responds back to the web client.

The data trail, which the web server interacts with, is a database that stores event data which date minimum a month back. This includes audit data and windmill status data.
If a user wants to see status of a windmill dating back to for instance 2 weeks ago the server looks up the data in the data trail and sends back the status of the windmill in question for the requested date. But if a user wants to see the current status of a windmill, the server sends a request to the windmill and the corresponding reply is sent back.
In order to have a registry of the behaviour of the windmills over a period of time, it is necessary to read windmill data periodically. This is done by the web server sending status requests regularly to windmills through the DMC application.

**Windmill**

A windmill in the TOE, besides hardware components, consists of the windmill-side of the DMC application and a Windmill Driver which are running on the windmill operating system. The DMC application takes care of receiving requests sent from users through the server to the windmills in question and vice versa. The DMC application processes received data from both the web server and the Windmill Driver such that it is represented and passed on correctly (e.g. right format of data).
The Windmill Driver collects data about the windmill and furthermore can on basis of the requests give instructions to change windmill attributes (such as rotor speed, gear, brakes, produced amount of electricity, etc.), ie. the Windmill Driver is the application that is in connection with the hardware related parts of the windmill and can control and read data from these.

### B.2.1 TOE Data Flows

The communication between TOE components is seen as a web client-web server and a web server-windmill interaction (see figure B.2).
Following data flows are specified for the TOE:

**User command requests**

Users that are allowed access to the TOE via the web interface can send requests concerning:

- Change of windmill attributes (allowed for operation engineers and administrators);
- View windmill attributes (allowed for all roles).

The web interface provides functionalities that allow users to make these requests. Furthermore, the web client can by itself send a "view" command to the web server in order to update the data on the screen with the newest information about windmills.

**User command replies**

As reply to user command requests the web server can, after processing the requests as described in previous section, respond with the following:

- Messages about the status of made requests[2];
- Windmill attributes.

The responses are received by the web client and shown on the web interface.

**Administrative command requests**

Since the administrators have the responsibility of administrating the TOE, it is possible for them to make requests concerning view or change of data in the web server or the windmills.
The administrators are hence able to fetch log information by requesting the data trail from the web server. The administrators could be interested in for instance reading who and when entities have tried to access services provided by the web server. This could be done in context with auditing of the server in order to carry out required security procedures in the TOE. Additionaly, administrators can make requests concerning role management issues, e.g. adding/removing users, changing/updating user information, and changing/updating user rights and privileges in the TOE.

**Administrative command replies**

Replies to the administrator requests include information about whether the requests are carried out successfully or not. Additionally the replies can contain requested data.

**Windmill status messages**

These messages consist of data concerning windmill operation. This could be a message from the server to a windmill asking for status or a message with actual windmill data from windmill to server.

[2]The status information could for instance be "Pending", "Changed", "Failed", or "Cancelled". There could be some additional information that can describe the status further more.

### B.2.2 TOE Devices

The devices that are used to access the TOE are computers and laptops. This has been decided because computers and laptops are common work stations for employees in corporate organisations. The computers and laptops used in the TOE are devices provided by the organisation, and which are set up with required configurations by administrators in order to obtain a controlled environment from which access to the TOE is established. If a user makes use of a computer that administrators do not have control of, it is possible that the computer is infected with various malware which could jeopardise the security of the TOE.

### B.2.3 TOE Roles

Another factor to consider is the roles users of the DMC system are assigned. Each role interacts in its own individual way with the system and thereby uphold individual rights and permissions when security is concerned.
Following roles are identified:

**Service technician -** The role of a service technician is to make sure that apparatus and instruments of the devices work correctly, ie. responsible in maintaining the physical security and has nothing to do with the IT functionality. In order to judge whether there is any malfunction in the system, it has to be possible for the technician to read (monitor) the status of the devices in the DMC system. The workplace of a technician in the Windmill DMC System is at the windmills out in the field.

**Operations engineer -** The operations engineer's job is to monitor and control the DMC system. Depending on regional divisions, number of operations engineers, workload, etc. rights and privileges to access data of the DMC system can vary from engineer to engineer. In other words operations engineers may monitor and control a subset of the system. Operations engineers are furthermore responsible for validating that both monitoring data and control data in the DMC system are correct. The operations engineer can perform the tasks from either the control centre or out in the field.

**Administrator -** The administrator is the one who is responsible for the overall functionality and security of the DMC system. The administrator of the DMC system owns rights and privileges to perform changes (installation and configuration) in order to maintain the functions and security

of the DMC system. In addition the administrator is responsible of user accounts, ie. creation of new user profiles with appropriate rights and privileges as well as maintenance of already existing user profiles.

The roles of the DMC system could be either static or dynamic, ie. the rights and privileges of a role can change over time or not. A dynamic model would be more suitable if frequent occurrence of promotion/degradation and thereby change in range of responsibilities among roles is present.
When possessing a role it is obviously clear that individuals that own that role are competent and trustworthy to carry out the work and responsibility that is demanded.

## B.3 TOE Security Environment

In this section assets, threat agents, threats, assumptions, and OSPs for the TOE security environment will be outlined.

### B.3.1 Assets

In regards to the purpose of the TOE:

*a)* Monitor and control devices in the system in a distributed way.

*b)* Keep record of changes in the system, ie. proof/evidence through the data trail.

the assets that need protection are

- the data trail,
- monitoring data,
- and control data.

Any malicious modification in these data jeopardises the security of the TOE. Below is listed an overall description of the possible critical points in the model, where the assets are most vulnerable:

**Servers -** The data trail, that resides on servers, is valuable to the TOE since it contains significant information in order to uphold the security of the TOE. Furthermore all data received and sent through the servers are also vulnerable. Servers (and thereby all the assets) are vulnerable to system breakdown, physical damage or attacks by malicious users or programs.

**Windmills -** Windmills are other potential weakness points in the TOE. It is expected that windmills send correct monitoring data when requested and upon receiving control data they act correspondingly. Within a windmill monitoring data may be read incorrectly due to malfunction of monitoring apparatus or data may be modified by unauthorised entities. Furthermore, received correct control data may be modified by malicious activity in the windmill.

**Input/output devices -** These devices are one of the critical points in the model because they might be infected with malicious programs that could harm the TOE when users gain access to the TOE through these devices.

**Connections -** Since the connections in the TOE are established over an insecure media, they are potential targets for attacks. Both monitoring data and control data are threatened by data loss, being read and/or modified.

### B.3.2 Threat Agents

The threat agents are divided into 2 groups: *internal attackers* and *external attackers*.

**Internal attackers** are entities within the company itself.

**External attackers** are correspondingly entities outside the company borders.

Hereinafter the term *attacker* will cover both groups of threat agents.
There could be several reasons for wanting to break into the TOE and gain access to valuable TOE data. Among these could be jealousy, competition, industrial espionage, revenge, or fun.
This leads to the observation that a threat agent can be characterised by the factors such as expertise, available resources, and motivation [14]. It is obvious that an entity that is involved of all three factors is of greater threat to the TOE than an entity with lack of one or more of the factors. Observations show that the strongest factor is motivation. An entity with high motivation and a given level of expertise and a set of resources is more likely to launch an attack compared to another entity that has lower motivation but the same expertise

and resources [14].
Having said that, the factors expertise and resources do not have so much impact on whether an entity launches an attack or not, ie.:

> *low expertise + resources + high motivation* $\approx$ *high expertise + resources + high motivation*

and

> *less resources + expertise + high motivation* $\approx$ *more resources + expertise + high motivation.*

### B.3.3 Threats

Potential threats are identified and listed below.

**T.MASQUERADE** $\diamond$ Unauthorised user or process pretends to be another entity in order to gain access to data or other TOE resources.

**T.UNAUTHORISED_ACCESS** $\diamond$ Mischievous users or programs may gain unauthorised access to data which they are not allowed to according to the TOE security policy.

**T.MODIFICATION** $\diamond$ Attackers may try to maliciously fiddle with protected data of the TOE.

**T.UNATTENDED_SESSION** $\diamond$ An attacker may gain unauthorised access to an unattended session.

**T.ACCIDENTAL_USER_ERROR** $\diamond$ Users may make accidental errors that could jeopardise the security of the TOE.

**T.DATA_TRANSMISSION** $\diamond$ An attacker may alter the transmission and thereby the confidentiality and the integrity of the data in the TOE.

**T.CRYPTO_LEAK** $\diamond$ Key data or other executable code associated with the cryptographic functionality, which intends to protect the data in the TOE system, may be viewed, modified or deleted by mischievous users or programs.

### B.3.4 Assumptions

The following descriptions identify the assumptions needed for the TOE to be securely operational.

**A.CORRECT_DEVELOPMENT** ◇ The development of the TOE, ie. design, implementation, and test, is assumed to be carried out correctly so it results in a TOE without flaws and errors that may lead to exploration by malicious users or programs.

**A.EXTERNAL_PARTY** ◇ Any external parties and products which the TOE relies upon are assumed trusted and fully functioning.

**A.NO_EVIL** ◇ It is assumed that administrators have no evil intentions and that they are appropriately trained to carry out their job correctly.

**A.PHYSICAL** ◇ The physical security of TOE is assumed provided in order to avoid physical loss or damage of the TOE due to external factors like fire, theft, natural catastrophes etc. Thus by this assumption the security of the data and the functionality of TOE are preserved.

### B.3.5 Organisational Security Policies (OSPs)

The OSPs are a set of rules, practices and procedures imposed by the organisation to address security needs. Following OSPs are identified:

**P.AUTHORISED_USERS** ◇ The TOE can only be accessed by authorised users.

**P.USER_PRIVILEGES** ◇ Users have different rights and privileges to access TOE data.

**P.ACCOUNTABILITY** ◇ Users that are authorised access to TOE data shall be held accountable for their actions within the TOE.

**P.CRYPTOGRAPHY** ◇ All cryptographic services used in the TOE must comply with the Federal Information Processing Standard Publication (FIPS PUB) 140-2 level 1.

**P.TRAIN** ◇ Authorised users of the TOE shall be trained appropriately in operating the TOE.

## B.4 Security Objectives

Following will contain an overview of the security objectives which aim at countering identified threats and/or comply with any OSPs and assumptions that were identified in section B.3. The rationale for these objectives can be found in section B.8.1.

### B.4.1 Security Objectives for the TOE

**O.UNIQUE_IA** ◇ The TSF shall ensure that unauthorised access to data in the TOE is not allowed. This shall be done by unique identification and authentication of entities trying to gain access to the TOE.

**O.DATA_INTEGRITY** ◇ Unauthorised modification, theft, or deletion of TOE data (user data and TSF data) shall be prevented.

**O.ACCOUNTABILITY_AND_AUDIT_RECORDS** ◇ The TSF shall provide individual accountability for audited events. The audit records shall record date and time of action and the identity of the entity responsible for the action.

**O.CRYPTO_FUNCTIONS** ◇ The TSF shall implement functions that comply with the Federal Information Processing Standard Publication (FIPS PUB) 140-2 level 1.

**O.ROLE_MANAGEMENT** ◇ The TSF shall provide a mechanism for administrators to control rights and privileges according to user roles.

**O.SESSION** ◇ The TSF shall provide mechanisms that lock sessions automatically when the activity in an open session has been idle in a predefined period of time. Furthermore it shall be possible for users to manually lock a session in order to avoid signing out. Users shall be able to unlock a session by re-authentication and just continue the session where it was left.

**O.BACK-UP** ◇ The TSF shall provide procedures for back-up of TOE data. The data trail must be recoverable at any time.

**O.VULNERABILITY_ANALYSIS** ◇ The TSF will undergo vulnerability analysis in order to verify that design, implementation and test of the TOE does not contain any flaws.

**O.CRYPTO_SECRECY** ◇ Key data or other executable code associated with the cryptographic functionality shall be kept secret.

**O.SELF_TEST** ◇ The TOE shall provide self-testing functionality for all TOE security functions which can detect security vulnerabilities in the form of flaws and intrusions.

### B.4.2 Security Objectives for the Environment

**OE.TRAIN** ◇ Training of administrators, operational engineers, and service technicians will be provided by the overall responsible of the TOE.

**OE.ISOLATION** ◇ Those responsible for the TOE shall provide isolation of physical parts of the TOE such that they are protected from physical damages, intrusion, and theft.

## B.5 Extended Components Definition

No extended components is included in this ST.

## B.6 Security Requirements

In this section all requirements to the TOE will be stated. This includes functional, assurance and TOE environment requirements. The corresponding rationale for these requirements can be found in section B.8.

### B.6.1 TOE Security Functional Requirements

This section provides functional requirements that must be satisfied by a PP-compliant TOE. These requirements consist of functional components from CC part 2. The components have been identified to fulfill the security objectives stated in previous section. The rationale behind identifying these components can be found in section B.8.2. Notice that selection and assignment identifications of some components are left to the ST authors. Table B.1 gives an overview of selected SFRs.

| SFR | Description |
|---|---|
| FAU_ARP.1 | Security alarms |
| FAU_GEN.1 | Audit data generation |
| FAU_GEN.2 | User identity association |
| FAU_SAA.1 | Potential violation analysis |
| FAU_SAR.1 | Audit review |
| FAU_SAR.2 | Restricted audit review |
| FAU_STG.1 | Protected audit trail storage |
| FCS_CKM.1 | Cryptographic key generation |
| FCS_CKM.2 | Cryptographic key distribution |
| FCS_CKM.4 | Cryptographic key destruction |
| FCS_COP.1 | Cryptographic operation |
| FDP_IFC.1 | Subset information flow control |
| FDP_IFF.1 | Simple security attributes |
| FDP_ITT.1 | Basic internal transfer protection |
| FDP_SDI.1 | Stored data integrity monitoring |
| FIA_UAU.2 | User authentication before any action |
| FIA_UAU.3 | Unforgeable authentication |
| FIA_UAU.6 | Re-authenticating |
| FIA_UID.2 | User identification before any action |
| FMT_MOF.1 | Management of security functions behaviour |
| FMT_MSA.1 | Management of security attributes |
| FMT_MSA.2 | Secure security attributes |
| FMT_MSA.3 | Static attribute initialisation |
| FMT_MTD.1 | Management of TSF data |
| FMT_SMF.1 | Specification of Management Functions |
| FMT_SMR.1 | Security roles |
| FPT_AMT.1 | Abstract machine testing |
| FPT_ITT.1 | Basic internal TSF data transfer protection |
| FPT_RCV.2 | Automated recovery |
| FPT_STM.1 | Reliable time stamps |
| FPT_TST.1 | TSF testing |
| FTA_SSL.1 | TSF-initiated session locking |
| FTA_SSL.2 | User-initiated locking |

Table B.1: Overview of identified SFRs.

#### B.6.1.1 Class FAU: Security audit

**FAU_ARP.1** **Security alarms**

FAU_ARP.1.1 The TSF shall take following actions[3] upon detection of a potential security violation:

*a) Show the alert visually;*

*b) Send e-mail to administrators;*

*c) Apply SNMP Traps;*

*d) Block traffic.*

**FAU_GEN.1** **Audit data generation**

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

- Start-up and shutdown of the audit functions;
- All auditable events for the *minimum*[4] level of audit; and
- *Start-up and shutdown of the server and windmill applications; and*
- *all user-initiated events.*

**Application note:** *Auditable events for the default level of audit include all minimum requirements and are identified in table B.2.*

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

- Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
- For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, *none*[5].

[3] [assignment: *list of actions*]
[4] [selection, choose one of: *minimum, basic, detailed, not specified*]
[5] [assignment: *other audit relevant information*]

| Functional Component | Auditable Event |
|---|---|
| FAU_ARP.1 | Actions taken due to potential security violations. |
| FAU_SAA.1 | Enabling and disabling of any of the analysis mechanisms. |
| | Automated responses performed by the tool. |
| FCS_CKM.1<br>FCS_CKM.2<br>FCS_CKM.4 | Success and failure of the activity. |
| FCS_COP.1 | Success and failure, and the type of cryptographic operation. |
| FDP_IFF.1 | Decisions to permit requested information flows. |
| FDP_ITT.1 | Successful transfers of user data, including identification of the protection method used. |
| FDP_SDI.1 | Successful attempts to check the integrity of user data, including an indication of the results of the check. |
| FIA_UAU.2 | Unsuccessful use of the authentication mechanism |
| FIA_UAU.3 | Detection of fraudulent authentication data. |
| FIA_UAU.6 | Failure of reauthentication. |
| FIA_UID.2 | Unsuccessful use of the user identification mechanism, including the user identity provided. |
| FMT_MSA.2 | All offered and rejected values for a security attribute. |
| FMT_SMF.1 | Use of the management functions. |
| FMT_SMR.1 | Modifications to the group of users that are part of a role. |
| FPT_ITT.1 | The detection of modification of TSF data. |
| FPT_RCV.1 | The fact that a failure or service discontinuity occurred. |
| | Resumption of the regular operation. |
| FPT_STM.1 | Changes to the time. |
| FTA_SSL.1 | Locking of an interactive session by the session locking mechanism. |
| FTA_SSL.2 | Successful unlocking of an interactive session. |

Table B.2: Additional auditable events for the minimal level of audit, from CC components.

**FAU_GEN.2** **User identity association**

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

**FAU_SAA.1** **Potential violation analysis**

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

- Accumulation or combination of *all in table B.2 specified auditable events*[6], known to indicate a potential security violation;
- *Start-up and shutdown of the audit functions.*[7]

**FAU_SAR.1** **Audit review**

FAU_SAR.1.1 The TSF shall provide *administrators*[8] with the capability to read *all audit information*[9] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

**FAU_SAR.2** **Restricted audit review**

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

**FAU_STG.1** **Protected audit trail storage**

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to *prevent*[10] unauthorised modifications to the stored audit records in the audit trail.

[6] [assignment: *subset of defined auditable events*]
[7] [assignment: *any other rules*]
[8] [assignment: *authorised users*]
[9] [assignment: *list of audit information*]
[10] [selection, choose one of: *prevent, detect*]

### B.6.1.2 Class FCS: Cryptographic support

**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 *Secure Hash Algorithm (SHA) as stated in the Secure Hash Standard (SHS)*[11] and specified cryptographic key sizes *dependent on which actual cipher suite is selected for the TOE to base its cryptographic operations upon*[12] that meet the following[13]:

*a) FIPS 140 level 1 or equivalent.*

**Application note:** The list of approved cryptographic algorithms for symmetric encryption of the data flow between TOE components are listed in FCS_COP.1.1.

**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 *RSA key exchange algorithm*[14] that meets the following[15]:

*a) FIPS 140 level 1 or equivalent.*

**FCS_CKM.4** **Cryptographic key destruction**

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with *any*[16] cryptographic key destruction method that meets the following[17]:

*a) FIPS 140 level 1 or equivalent.*

**FCS_COP.1** **Cryptographic operation**

FCS_COP.1.1(2) The TSF shall perform *encryption of the data flow and mutual authentication between the components of the TOE*[18] in accordance with *any of the following TLS cipher suites*[19]:

[11] [assignment: *cryptographic key generation algorithm*]
[12] [assignment: *cryptographic key sizes*]
[13] [assignment: *list of standards*]
[14] [assignment: *cryptographic key distribution method*]
[15] [assignment: *list of standards*]
[16] [assignment: *cryptographic key destruction method*]
[17] [assignment: *list of standards*]
[18] [assignment: *list of cryptographic operations*]
[19] [assignment: *cryptographic algorithm*]

*a) TLS_RSA_WITH_AES_128_CBC_SHA*

*b) TLS_RSA_WITH_AES_256_CBC_SHA*

and cryptographic key sizes of *128 or 256 bit for AES, and a minimum of 1024 bit for RSA* that meet the following[20]:

*a) FIPS 140 level 1 or equivalent.*

[20] [assignment: *list of standards*]

### B.6.1.3 Class FDP: User data protection

**FDP_IFC.1 Subset information flow control**

FDP_IFC.1.1 The TSF shall enforce the *DATA_FLOW_SFP*[21] on *data flow* between *the web clients and the web server*, and *the web server and the windmills*[22].

**Application note:** *The data flows are described in section B.2.1.*

**FDP_IFF.1 Simple security attributes**

FDP_IFF.1.1 The TSF shall enforce the *DATA_FLOW_SFP*[23] based on the following types of subject and information security attributes[24]:

**Subject security attributes:** *a) Type of input/output devices which the information flows between;*
*b) Roles of the entities that cause the information to flow or roles of the entities that act as recipients of the information.*

**Information security attributes:** *a) Type of information;*
*b) Sensitivity of the information.*

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold[25]:

*a) The input/output devices are authenticated before any data flow is allowed between them;*

*b) Entities own the rights to carry out actions that imply data flow;*

*c) Data is encrypted such that the data flow has to be carried out using one of the cipher suites TLS_RSA_WITH_AES_128_CBC_SHA or TLS_RSA_WITH_AES_256_CBC_SHA in SSL 3.1/TLS 1.0;*

[21] [assignment: *information flow control SFP*]
[22] [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]
[23] [assignment: *information flow control SFP*]
[24] [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*]
[25] [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*]

FDP_IFF.1.3 The TSF shall enforce *no other additional information flow control SFP rules*[26].

FDP_IFF.1.4 The TSF shall provide the following[27]:

- *No additional SFP capabilities.*

FDP_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules[28]:

- *None*

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules[29]:

- *None*

**FDP_ITT.1 Basic internal transfer protection**

FDP_ITT.1.1 The TSF shall enforce the *DATA_FLOW_SFP*[30] to prevent the *disclosure* and *modification*[31] of user data when it is transmitted between physically-separated parts of the TOE.

**FDP_SDI.1 Stored data integrity monitoring**

FDP_SDI.1.1 The TSF shall monitor user data stored in containers controlled by the TSF for *data that is changed or lost because of errors in the hardware*[32] on all objects, based on the following attributes[33]:

*a) The contents of the data;*

*b) The creator of the data;*

*c) Date of creation and modification of the data;*

*d) Read/write permissions of data.*

[26] [assignment: *additional information flow control SFP rules*]
[27] [assignment: *list of additional SFP capabilities*]
[28] [assignment: *rules, based on security attributes, that explicitly authorise information flows*]
[29] [assignment: *rules, based on security attributes, that explicitly deny information flows*]
[30] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
[31] [selection: *disclosure, modification, loss of use*]
[32] [assignment: *integrity errors*]
[33] [assignment: *user data attributes*]

### B.6.1.4 Class FIA: Identification and authentication

**FIA_UID.2** **User identification before any action**

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

**FIA_UAU.2** **User authentication before any action**

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

**FIA_UAU.3** **Unforgeable authentication**

FIA_UAU.3.1 The TSF shall *prevention*[34] use of authentication data that has been forged by any user of the TSF.

FIA_UAU.3.2 The TSF shall *prevention*[35] use of authentication data that has been copied from any other user of the TSF.

**FIA_UAU.6** **Re-authenticating**

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions[36]:

*a) A session has been locked and the user wants to unlock the session.*

### B.6.1.5 Class FMT: Security management

**FMT_MOF.1** **Management of security functions behaviour**

FMT_MOF.1.1 The TSF shall restrict the ability to *determine the behaviour* and *modify the behaviour*[37] the functions[38]:

a) *Functions implementing creation and recovery of back-ups;*

[34] [selection: *detect, prevent*]
[35] [selection: *detect, prevent*]
[36] [assignment: *list of conditions under which re-authentication is required*]
[37] [selection: *determine the behaviour of, disable, enable, modify the behaviour of*]
[38] [assignment: *list of functions*

b) *Functions implementing role management, including administration and maintenance of delegated roles;*

c) *Functions implementing routines for identifying events that have to be audited and administration and maintenance of audit records;*

d) *Functions implementing methods for identification and authentication of users;*

e) *Functions implementing and maintaining access control methods;*

f) *Functions implementing methods for locking sessions;*

g) *Functions implementing secure procedures for data transfer;*

h) *Functions implementing procedures for assuring physical security and its maintenance;*

i) *Functions implementing methods for self test and analysis of results from the testing;*

j) *Functions implementing timers and clock synchronisation;*

k) *Functions for managing any cryptography related issues;*

l) *No other additional manageable functions*[39].

to *the administrators*[40].

**FMT_MSA.1 Management of security attributes**

FMT_MSA.1.1 The TSF shall enforce the *DATA_FLOW_CONTROL_SFP*[41] to restrict the ability to *modify*[42] the security attributes *that are defined in the DATA_FLOW_CONTROL_SFP*[43] to *administrators*[44].

**FMT_MSA.2 Secure security attributes**

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

[39] [Assignment: *additional manageable functions*]
[40] [assignment: *the authorised identified roles*
[41] [assignment: *access control SFP, information flow control SFP*]
[42] [selection: *change_default, query, modify, delete*, [assignment: *other operations*]]
[43] [assignment: *list of security attributes*]
[44] [assignment: *the authorised identified roles*]

**FMT_MSA.3 Static attribute initialisation**

FMT_MSA.3.1 The TSF shall enforce the *DATA_FLOW_CONTROL_SFP*[45] to provide *restrictive*[46] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the *administrators*[47] to specify alternative initial values to override the default values when an object or information is created.

**FMT_MTD.1 Management of TSF data**

FMT_MTD.1.1 The TSF shall restrict the ability to *modify*, *query*, and *delete* [48] the[49]:

a) *Data trail;*

b) *Identification and authentication data;*

c) *Cryptographic algorithms and keys;*

d) *Audit records; and*

e) [assignment: *additional TSF data*]

to *the administrators*[50].

**FMT_SMF.1 Specification of Management Functions**

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions[51] :

a) *Functions to create and recover back-ups;*

b) *Functions to administrate and maintain delegated roles;*

c) *Functions to maintain audit records and to identify events that have to be audited and administrated;*

d) *Functions to identify and authenticate users;*

e) *Functions to protect data by using access control methods;*

[45] [assignment: *access control SFP, information flow control SFP*]
[46] [selection, choose one of: *restrictive, permissive,* [assignment: *other property*]]
[47] [assignment: *the authorised identified roles*]
[48] [selection: *change_default, query, modify, delete, clear,* [assignment: *other operations*]]
[49] [assignment: *list of TSF data*]
[50] [assignment: *the authorised identified roles*]
[51] [assignment: *list of management functions to be provided by the TSF*]

f) *Functions setting up and maintaining session locking attributes;*

g) *Functions that secure procedures for data transfer;*

h) *Functions assuring physical security and its maintenance;*

i) *Functions for self testing and analysing results from the testing;*

j) *Functions to synchronise timers and clock;*

k) *Functions that manage cryptography related issues;*

l) *No other additional manageable functions*[52].

**FMT_SMR.1 Security roles**

FMT_SMR.1.1 The TSF shall maintain the roles[53]:

a) *Service technician;*

b) *Operations engineer; and*

c) *administrator.*

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

#### B.6.1.6 Class FPT: Protection of the TSF

**FPT_AMT.1 Abstract machine testing**

FPT_AMT.1.1 The TSF shall run a suite of tests *during initial start-up, periodically during normal operation, and at the request of an authorised user*[54] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

[52] [Assignment: *additional manageable functions*]

[53] [assignment: *the authorised identified roles*]

[54] [selection: *during initial start-up, periodically during normal operation, at the request of an authorised user,* [assignment: *other conditions*]]

**FPT_ITT.1** **Basic internal TSF data transfer protection**

FPT_ITT.1.1 The TSF shall protect TSF data from *disclosure and modification*[55] when it is transmitted between separate parts of the TOE.

**FPT_STM.1** **Reliable time stamps**

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

**FPT_TST.1** **TSF testing**

FPT_TST.1.1 The TSF shall run a suite of self tests *during initial start-up, periodically during normal operation, and at the request of an authorised user*[56] to demonstrate the correct operation of *the TSF*[57].

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of *TSF data*[58].

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

**FPT_RCV.2** **Automated recovery**

FPT_RCV.2.1 When automated recovery from *power failure* and *system failures*[59] is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.2.2 For *system failures*[60], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

[55] [selection: *disclosure, modification*]
[56] [selection: *during initial start-up, periodically during normal operation, and at the request of the authorised user, at the conditions* [assignment: *conditions under which self test should occur*]]
[57] [selection: [assignment: *parts of TSF*], *the TSF*]
[58] [selection: [assignment: *parts of TSF*], *TSF data*]
[59] [assignment: *list of failures/service discontinuities*]
[60] [assignment: *list of failures/service discontinuities*]

#### B.6.1.7 Class FTA: TOE access

**FTA_SSL.1 TSF-initiated session locking**

FTA_SSL.1.1 The TSF shall lock an interactive session after *15 minutes*[61] by:

- clearing or overwriting display devices, making the current contents unreadable;
- disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 The TSF shall require the following events to occur prior to unlocking the session[62]:

a) *Re-authentication of the user that wants to unlock the session.*

[61] [assignment: *time interval of user inactivity*]
[62] [assignment: *events to occur*]

### B.6.2 TOE Security Assurance Requirements

The assurance level of this TOE is EAL3 compliant. Following will list the TOE security assurance requirements. The EAL consists of assurance components that each meet an assurance requirement. The components are taken from the CC part 3.

| *Assurance Class* | *Assurance components/SARs* | |
|---|---|---|
| ADV:Development | ADV_ARC.1 | Security architecture description |
| | ADV_FSP.3 | Functional specification with complete summary |
| | ADV_TDS.2 | Architectural design |
| AGD:Guidance documents | AGD_OPE.1 | Operational user guidance |
| | AGD_PRE.1 | Preparative procedures |
| ALC:Life-cycle support | ALC_CMC.3 | Authorisation controls |
| | ALC_CMS.3 | Implementation representation CM coverage |
| | ALC_DEL.1 | Delivery procedures |
| | ALC_DVS.1 | Identification of security measures |
| | ALC_LCD.1 | Developer defined life-cycle model |
| ASE:Security Target evaluation | ASE_CCL.1 | Conformance claims |
| | ASE_ECD.1 | Extended components definition |
| | ASE_INT.1 | ST introduction |
| | ASE_OBJ.2 | Security objectives |
| | ASE_REQ.2 | Derived security requirements |
| | ASE_SPD.1 | Security problem definition |
| | ASE_TSS.1 | TOE summary specification |
| ATE:Tests | ATE_COV.2 | Analysis of coverage |
| | ATE_DPT.1 | Testing: basic design |
| | ATE_FUN.1 | Functional testing |
| | ATE_IND.2 | Independent testing - sample |
| AVA:Vulnerability assessment | AVA_VAN.2 | Vulnerability analysis |

Table B.3: Security assurance components/requirements in EAL3[7].

#### B.6.2.1 Class ADV: Development

**ADV_ARC.1 Security architecture description**

**Developer action elements:**

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

**Content and presentation elements:**

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialisation process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

**Evaluator action elements:**

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

### ADV_FSP.3 Functional specification with complete summary

**Developer action elements:**

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

ADV_FSP.3.2D The developer shall provide a tracing from the functional specification to the SFRs.

**Content and presentation elements:**

ADV_FSP.3.1C The functional specification shall completely represent the TSF.

ADV_FSP.3.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.3.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.3.4C For SFR-enforcing TSFIs, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.

ADV_FSP.3.5C For SFR-enforcing TSFIs, the functional specification shall describe direct error messages resulting from security enforcing effects and exceptions associated with invocation of the TSFI.

ADV_FSP.3.6C The functional specification shall summarise the non-SFR-enforcing actions associated with each TSFI.

ADV_FSP.3.7C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

**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 SFRs.

**ADV_TDS.2 Architectural design**

**Developer action elements:**

ADV_TDS.2.1D The developer shall provide the design of the TOE.

ADV_TDS.2.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

**Content and presentation elements:**

ADV_TDS.2.1C The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.2.2C The design shall identify all subsystems of the TSF.

| | |
|---|---|
| ADV_TDS.2.3C | The design shall describe the behaviour of each SFR non-interfering subsystem of the TSF in detail sufficient to determine that it is SFR noninterfering. |
| ADV_TDS.2.4C | The design shall describe the SFR-enforcing behaviour of the SFR-enforcing subsystems. |
| ADV_TDS.2.5C | The design shall summarise the non-SFR-enforcing behaviour of the SFR-enforcing subsystems. |
| ADV_TDS.2.6C | The design shall summarise the behaviour of the SFR-supporting subsystems. |
| ADV_TDS.2.7C | The design shall provide a description of the interactions among all subsystems of the TSF. |
| ADV_TDS.2.8C | The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it. |

**Evaluator action elements:**

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

#### B.6.2.2 Class AGD: Guidance documents

**AGD_OPE.1 Operational user guidance**

**Developer action elements:**

| | |
|---|---|
| AGD_OPE.1.1D | The developer shall provide operational user guidance. |

**Content and presentation elements:**

| | |
|---|---|
| AGD_OPE.1.1C | The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings. |
| AGD_OPE.1.2C | The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner. |

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance 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.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

**Evaluator action elements:**

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

### AGD_PRE.1 Preparative procedures

**Developer action elements:**

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

**Content and presentation elements:**

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

**Evaluator action elements:**

| | |
|---|---|
| AGD_PRE.1.1E | The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. |
| AGD_PRE.1.2E | The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation. |

#### B.6.2.3 Class ALC: Life-cycle support

**ALC_CMC.3 Authorisation controls**

**Developer action elements:**

| | |
|---|---|
| ALC_CMC.3.1D | The developer shall provide the TOE and a reference for the TOE. |
| ALC_CMC.3.2D | The developer shall provide the CM documentation. |
| ALC_CMC.3.3D | The developer shall use a CM system. |

**Content and presentation elements:**

| | |
|---|---|
| ALC_CMC.3.1C | The TOE shall be labelled with its unique reference. |
| ALC_CMC.3.2C | The CM documentation shall describe the method used to uniquely identify the configuration items. |
| ALC_CMC.3.3C | The CM system shall uniquely identify all configuration items. |
| ALC_CMC.3.4C | The CM system shall provide measures such that only authorised changes are made to the configuration items. |
| ALC_CMC.3.5C | The CM documentation shall include a CM plan. |
| ALC_CMC.3.6C | The CM plan shall describe how the CM system is used for the development of the TOE. |
| ALC_CMC.3.7C | The evidence shall demonstrate that all configuration items are being maintained under the CM system. |
| ALC_CMC.3.8C | The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan. |

**Evaluator action elements:**

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

**ALC_CMS.3 Implementation representation CM coverage**

**Developer action elements:**

ALC_CMS.3.1D The developer shall provide a configuration list for the TOE.

**Content and presentation elements:**

ALC_CMS.3.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; and the implementation representation.

ALC_CMS.3.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.3.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

**Evaluator action elements:**

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

**ALC_DEL.1 Delivery procedures**

**Developer action elements:**

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

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

**Content and presentation elements:**

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

**Evaluator action elements:**

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

**ALC_DVS.1 Identification of security measures**

**Developer action elements:**

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

**Content and presentation 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.

**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.

**ALC_LCD.1 Developer defined life-cycle model**

**Developer action elements:**

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

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

**Content and presentation elements:**

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

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

**Evaluator action elements:**

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

#### B.6.2.4 Class ASE: Security Target evaluation

**ASE_CCL.1 Conformance claims**

**Developer action elements:**

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

**Content and presentation elements:**

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

**Evaluator action elements:**

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

**ASE_ECD.1 Extended components definition**

**Developer action elements:**

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

**Content and presentation elements:**

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or non-conformance to these elements can be demonstrated.

**Evaluator action elements:**

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

ASE_ECD.1.2E The evaluator shall confirm that no extended component can be clearly expressed using existing components.

**ASE_INT.1 ST introduction**

**Developer action elements:**

ASE_INT.1.1D The developer shall provide an ST introduction.

**Content and presentation elements:**

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall identify the TOE.

ASE_INT.1.4C The TOE overview shall summarise the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

**Evaluator action elements:**

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

ASE_INT.1.2E The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

**ASE_OBJ.2 Security objectives**

**Developer action elements:**

ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objectives rationale.

**Content and presentation elements:**

ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.

ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

**Evaluator action elements:**

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

**ASE_REQ.2 Derived security requirements**

**Developer action elements:**

ASE_REQ.2.1D The developer shall provide a statement of security requirements.

ASE_REQ.2.2D The developer shall provide a security requirements rationale.

**Content and presentation elements:**

ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.2.4C All operations shall be performed correctly.

ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.

ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

**Evaluator action elements:**

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

**ASE_SPD.1 Security problem definition**

**Developer action elements:**

ASE_SPD.1.1D The developer shall provide a security problem definition.

**Content and presentation elements:**

ASE_SPD.1.1C The security problem definition shall describe the threats.

ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C The security problem definition shall describe the OSPs.

ASE_SPD.1.4C The security problem definition shall describe the assumptions about the operational environment of the TOE.

**Evaluator action elements:**

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

**ASE_TSS.1 TOE summary specification**

**Developer action elements:**

ASE_TSS.1.1D The developer shall provide a TOE summary specification.

**Content and presentation elements:**

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

**Evaluator action elements:**

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

ASE_TSS.1.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

#### B.6.2.5 Class ATE: Tests

**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 elements:**

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

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

**Evaluator action elements:**

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

**ATE_DPT.1 Testing: basic design**

**Developer action elements:**

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

**Content and presentation elements:**

ATE_DPT.1.1C The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems in the TOE design.

ATE_DPT.1.2C The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

**Evaluator action elements:**

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

**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 elements:**

| | |
|---|---|
| ATE_FUN.1.1C | The test documentation shall consist of test plans, expected test results and actual test results. |
| ATE_FUN.1.2C | The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests. |
| ATE_FUN.1.3C | The expected test results shall show the anticipated outputs from a successful execution of the tests. |
| ATE_FUN.1.4C | The actual test results shall be consistent with the expected test results. |

**Evaluator action elements:**

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

**ATE_IND.2 Independent testing - sample**

**Developer action elements:**

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

**Content and presentation elements:**

| | |
|---|---|
| ATE_IND.2.1C | The TOE shall be suitable for testing. |
| ATE_IND.2.2C | The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF. |

**Evaluator action elements:**

| | |
|---|---|
| ATE_IND.2.1E | The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. |
| ATE_IND.2.2E | The evaluator shall execute a sample of tests in the test documentation to verify the developer test results. |
| ATE_IND.2.3E | The evaluator shall test a subset of the TSF interfaces to confirm that the TSF operates as specified. |

#### B.6.2.6 Class AVA: Vulnerability assessment

**AVA_VAN.2 Vulnerability analysis**

**Developer action elements:**

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

**Content and presentation elements:**

AVA_VAN.2.1C The TOE shall be suitable for testing.

**Evaluator action elements:**

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

AVA_VAN.2.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3E The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

### B.6.3 TOE Environment Requirements

In this section the requirements for the TOE environment will be stated including the thoughts and rationale behind them.

**OE.TRAIN**

In order even to be considered for operating the TOE, a user must have assigned a role, identified in section A.2.5. This means that there must be some procedure for evaluation if a user can be assigned the role. These procedures should in principal be present in the assurance requirement called AGD_OPE.1 - Operational user guidance. It should specifically be stated how training shall be carried out in terms of which qualification tests users should go through, which certificates they should acquire, and which courses they should attend before they are assigned a role.

**OE.ISOLATION**
It was previously assumed (A.PHYSICAL) that physical protection was provided as a general assumption. Reasonable precautions like safety rooms, barriers and fences to sensitive parts of the TOE (see section A.3.3) are expected to be in order.
Although physical protection of the TSF is addressed in the "TSF physical protection (FPT_PHP)" family, it does not state any measures for encountering the actual damage or theft of parts. The components in the family only detect when/if something physical to parts happen and specify what should be done. This is not sufficient to fulfil the objective. But together with the stated general assumption the OE is satisfied to satisfactory extent.

It is important to keep in mind that the specified requirements for the operational environment of the TOE are guidance lines to reach a reasonable level of security in relation to cost and other resources available. Furthermore, it must be noticed that it is not possible to measure or test whether or how much the requirements for the environment are followed.

## B.7 TOE Summary Specification

The TOE summary specification outlines the security functions, security functional policies, and assurance measures of the TOE that meet the TOE security requirements.

### B.7.1 TOE Security Functions

The security functions that are identified in the SFR FMT_MOF.1 will be described in this section.

**F.BACK_UP** ◇ Within the discussed TOE, it is the management that is in charge of back-ups. The data to be backed-up is the TSF data and user data (ie. the data trail). The approach to be applied is as follows:

*a)* Every midnight a full back-up is made of all TSF data (including data trail) by copying the data into an external storage media.

*b)* During the day every change in TSF data and data trail is likewise stored in an external storage media.

The procedure of taking back-up of TOE data is an automated one. Back-up data is physically located a secure place that is different from the location where the web server is placed.

**F.ROLE** ◇ The TSF supports the roles of:

- Service technician,
- operating engineer, and
- administrator.

**F.AUDIT** ◇ The TSF generates an audit record of the following events, and associates each event with the user that caused the event:

- Start-up and shutdown of the audit functions;
- *All auditable events specified in table 6.4*;
- *Start-up and shutdown of the server and windmill applications; and*
- *all user-initiated events.*

The TSF records the following information for each audit event, where applicable:

- Date and time of the event;
- Type of event;
- Identity of the entity that caused the event; and
- success or failure of the event.

**F.AUTH** ◇ The TOE provides methods for identification and authentication of users before access to the TOE is granted.

**F.CRYPTOGRAPHY** ◇ This security function assures that cryptographic operations follow the FIPS 140 level 1 guidelines. The TSF implements the SSL 3.1/TLS 1.0 protocol for data communication between the components in the TOE.

**F.SCAN** ◇ The TSF provides methods for scanning the TSF for flaws and integrity errors. The tests are carried out under following conditions:

*a)* During initial start-up;

*b)* Periodically during normal operation; and

*c)* at the request of an authorised user.

**F.MANAGEMENT** ◇ Management functions listed in FMT_SMF.1 must be manageable for administrators.

## B.7.2 Security Functional Policies

In the following any security functional policies will be stated.

### B.7.2.1 SFP for Data Flow Control (DATA_FLOW_SFP)

The data flows identified for the TOE (see section B.2.1) will be enforced the rules stated in this SFP based upon the RBAC paradigm:

*a)* The input/output devices are authenticated before any data flow is allowed between them;

*b)* Entities own the rights to carry out actions that imply data flow (e.g. a service technician is not able to create a data flow for changing windmill attributes);

*c)* Data is encrypted such that the data flow has to be carried out using one of the cipher suites TLS_RSA_WITH_AES_128_CBC_SHA or TLS_RSA_WITH_AES_256_CBC_SHA in SSL 3.1/TLS 1.0.

Administrators have the responsibility of management of the information data flow control, ie. the administrators manage which entities have rights to carry out which data flow between the TOE components.

## B.7.3 Assurance Measures

Following states the assurance measures identified in meeting the assurance requirements. The rationale behind this specification can be found in section B.8.3.2.

**M.ARCH** ◇ The developer will provide a security architecture description of the TSF. This includes a description of the interactions among all subsystems of the TSF.

**M.SPEC** ◇ A high-level design and functional specification of the TOE will be provided by the developer for the evaluation. The documents describe the TOE security functionality, subsystems, and interfaces.

**M.DOCS**◇ The developer will provide operational user guidance which contains guidelines for configuration and handling of the TOE. This includes documents for preparation and operation of the TOE.

**M.SETUP** ◇ The developer will provide guidance documents describing secure receiving, installation, and configuration of the TOE.

**M.AUTH_CTRL**◇ The TOE is developed and maintained using a system to ensure only authorised changes are implemented in the evaluated version of the TOE, ie. a CM system is provided in the TOE. This system is documented. A list of all TOE documentation and all configuration items required to create the TOE is maintained.

**M.ID** ◇ A unique version identifier for the TOE will be available for the user.

**M.DELIVER** ◇ The developer will provide a controlled process and procedures whereby the developer will deploy the TOE for the customer. The process and procedures are documented.

**M.SECMEASURES** ◇ The developer will produce a development security documentation that describes 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.

**M.LIFE-CYCLE** ◇ The developer will provide a life-cycle definition documentation which will be used in the development and maintenance of the TOE.

**M.CLAIMS** ◇ The developer will provide a CC conformance claim and a corresponding conformance claim rationale.

**M.EXCOMP** ◇ The developer will provide a statement of security requirements and will provide definition of any extended components.

**M.INTRO** ◇ The developer will provide an ST introduction.

**M.OBJ** ◇ The developer will provide a statement of security objectives that describe the security objectives for the operational environment. Furthermore a rationale for the objectives will be included.

**M.REQ** ◇ The developer will provide a statement of security requirements and a corresponding security requirements rationale.

**M.SPD** ◇ The developer will provide a security problem definition in form of a description and specification of the TOE security environment (ie. assets, threat agents, threats, assumptions, and OSPs).

**M.TSS** ◇ The developer will provide a TOE summary specification which defines security functions, security measures, and any security functional policies.

**M.TEST** The developer will provide an analysis of the test coverage and the depth of testing. The developer will test the TSF and document the results. Furthermore the developer will provide the evaluator with materials necessary for the efficient reproduction of developer tests.

## B.8 Rationale

This section of the ST provides the rationale for the security objectives, security requirements (SFRs and SARs), TOE summary specification, and the conformance claim.

### B.8.1 Security Objectives Rationale

The following illustrates the mapping of security objectives to identified threats, assumptions, and policies. This mapping is sketched in table B.4. The table also illustrates that all identified threats, assumptions, and policies are covered by at least one security objective.

#### B.8.1.1 Rationale to counter Threats

Following rationale demonstrates how the objectives counter the threats.

**T.MASQUERADE**
This threat is primarily countered by the O.UNIQUE_IA, O.DATA_INTEGRITY and O.ACCOUNTABILITY_AND_AUDIT objectives.

O.UNIQUE_IA provides means for unique identification and authentication of users before they are granted access to the TOE services. It is clear that by ensuring this you are aware of who has access to the TOE and know exactly that it is authorised entities that gain access and thereby prevent masquerade from occurring.

| | O.UNIQUE_IA | O.DATA_INTEGRITY | O.ACCOUNTABILITY_AND_AUDIT | O.CRYPTO_FUNCTIONS | O.ROLE_MANAGEMENT | O.SESSION | O.BACK-UP | O.VULNERABILITY_ANALYSIS | O.CRYPTO_SECRECY | O.SELF_TEST | OE.TRAIN | OE.ISOLATION |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T.MASQUERADE | × | × | × | | | | | | | | | |
| T.UNAUTHORISED_ACCESS | × | × | × | | × | | | | | | | |
| T.MODIFICATION | × | × | × | × | | | × | | | | | |
| T.UNATTENDED_SESSION | | | | | | × | | | | | × | |
| T.ACCIDENTAL_USER_ERROR | | | × | | | | × | | | | × | |
| T.DATA_TRANSMISSION | | × | | × | | | | | | | | |
| T.CRYPTO_LEAK | | | | | | | | | × | | | |
| A.CORRECT_DEVELOPMENT | | | | | | | | × | | | | |
| A.EXTERNAL_PARTY | × | × | | × | | | | | × | | × | |
| A.NO_EVIL | | | | | | | | | | | × | |
| A.PHYSICAL | | | | | | | × | | | | | × |
| P.AUTHORISED_USERS | × | × | × | | × | | | | | | | |
| P.USER_PRIVILEGES | | | | | × | | | | | | | |
| P.ACCOUNTABILITY | × | | × | | × | | | | | | | |
| P.CRYPTOGRAPHY | | | | × | | | | | × | | | |
| P.TRAIN | | | | | | | | | | | × | |

Table B.4: Objectives related to the TOE security environment.

O.DATA_INTEGRITY covers another aspect of the threat. It states that unauthorised modification, theft, or deletion of TOE data (user data and TSF data) shall be prevented. The fact that no unknown entity can perform any malicious activity on data in the TOE contributes in preventing T.MASQUERADE. Data integrity of TOE data is vital in order to keep the TOE reliable and trustworthy.

O.ACCOUNTABILITY_AND_AUDIT is another objective contributing in meeting the T.MASQUERADE threat. By providing individual accountability for audited events it is possible to keep track of user actions in the TOE and appropriate precautions can be taken if suspicious activities are recorded.

O.SELF_TEST is meant as a last option in detecting a masquerade but is not a preventing measure. The objective can detect malicious activity so precautions can be taken in order to protect the TOE against future similar attacks and/or limit the damages.

**T.UNAUTHORISED_ACCESS**

This threat is countered by the O.UNIQUE_IA, O.DATA_INTEGRITY, O.ACCOUNTABILITY_AND_AUDIT, and O.ROLE_MANAGEMENT objectives.

Exactly same reasoning as described for T.MASQUERADE can be applied to show that the O.UNIQUE_IA, O.DATA_INTEGRITY, O.ACCOUNTABILITY_AND_AUDIT, and O.SELF_TEST objectives also counter this threat.

The O.ROLE_MANAGEMENT objective comes into play when roles have to be administrated. This includes rights and privileges. Which resources of the TOE that should be accessible to which roles in the TOE must constantly be revised. Furthermore, in a company there will constantly be new staff being employed and old being fired and areas of responsibility might also be changed at times. So management of roles is a necessity in order to meet the unauthorised access threat.

**T.MODIFICATION**

This threat is encountered by following objectives: O.UNIQUE_IA, O.DATA_INTEGRITY, O.ACCOUNTABILITY_AND_AUDIT, O.CRYPTO_FUNCTIONS, and O.BACK-UP.
If a malicious modification of data has occurred the O.UNIQUE_IA objective can contribute in identifying which entity has been causing this.

O.DATA_INTEGRITY covers another aspect of the threat concerned. It states that unauthorised modification, theft, or deletion of TOE data (user

data and TSF data) shall be prevented. The fact that no unknown entity can perform any malicious activity on data in the TOE contributes in countering the threat.

O.ACCOUNTABILITY_AND_AUDIT provides means for individual accountability for audited events and thereby makes it possible to keep track of user actions in the TOE. Together with the O.UNIQUE_IA objective it enables detection of malicious modification of TOE data by going through the audit records.

The objective O.CRYPTO_FUNCTIONS provides cryptographic measures that preserve integrity and confidentiality of TOE data. Thus, it states that the TSF shall implement approved cryptographic algorithms. Encryption of data can prevent data from being viewed and modified.

Back-up of data is enforced by O.BACK-UP. This is found relevant to the modification threat because if any unauthorised modification should happen it is possible to recover previous states of data. Especially the data trail should be recoverable at any time.

Same reasoning as before can be applied for the O.SELF_TEST objective countering this threat.

**T.UNATTENDED_SESSION**

Unattended sessions are mainly covered by the objective O.SESSION. Besides this objective, the OE.TRAIN also has some influence on this threat.

O.SESSION provides demands for unattended idle sessions. The TSF shall provide mechanisms for locking user sessions automatically after a given idle time. Furthermore, the objective makes it possible for users to manually lock sessions and return to the session through re-authentication.

The environment objective OE.TRAIN also plays a part in securing the TOE against this threat. Appropriate training of users of the TOE include making them aware of possible risks in leaving sessions unattended which can minimize the threat.

Same reasoning as before can be applied for the O.SELF_TEST objective countering this threat.

The OE.ISOLATION environment objective addresses the threat from another perspective. It counters the T.UNATTENDED_SESSION threat by physically isolate the devices on which the sessions are running. It is only possible to isolate the control centre and thereby prevent unwanted

people to gain access to the control centre. So unattended sessions can be secured against in the control centre through isolation but this is not the case out in the field. It is not possible to isolate the whole world. This is why the isolation environment objective only partly counters the T.UNATTENDED_SESSION threat.

**T.ACCIDENTAL_USER_ERROR**

This threat can be met by the security objectives O.ACCOUNTABILITY-_AND_AUDIT, O.BACK-UP and OE.TRAIN.

The rationale behind the O.ACCOUNTABILITY_AND_AUDIT and O.BACK-UP objectives for meeting this threat is quite obvious. If any accidental user error is detected in the audit records or by users them selves, it is possible to investigate what has gone wrong through O.ACCOUNTABILITY_AND_AUDIT in audit records and eventually recover the TOE into a previous state, through the O.BACK-UP objective.

OE.TRAIN demands that users are aware of how to use the TOE correctly. Training procedures should make users competent and responsible when they obtain an access role in the TOE as stated in section B.2.3. Thus, the objective contributes in restricting the threat from causing damages to the TOE by minimizing accidental user errors through appropriate training.

The O.SELF_TEST objective partly covers the threat too by providing means for detecting an accidental user error testing and scanning the TOE.

**T.DATA_TRANSMISSION**

Transmission of data is a weakness point in the TOE. Therefore appropriate measures have to be applied in order to protect the data in transfer. This is mainly obtained through data integrity and data confidentiality. Thus, the objectives O.DATA_INTEGRITY and O.CRYPTO_FUNCTIONS counter this threat.
O.DATA_INTEGRITY ensures that unauthorised modification, theft, or deletion of TOE data shall be prevented. This also goes in regards to the connections over which the data is sent. This could be done through CRC (cyclic redundancy check) data check. The fact that no unknown entity can perform malicious activity on data in the connections of the TOE contributes in preventing T.DATA_TRANS-MISSION.

O.CRYPTO_FUNCTIONS provides countermeasures against this threat by encrypting data such that the data can not be read or modified by unauthorised entities. It could be in form of SSL encryption or using digital signatures which are typical cryptographic functions. The objective as such aims at protecting data not only within the TOE itself but also

data transmission connections and thus is related to this threat.

Same reasoning as before can be applied for the O.SELF_TEST objective countering this threat.

**T.CRYPTO_LEAK**
This threat is so specific that only one objective is identified to counter this threat. This is the O.CRYPTO_SECRECY security objective.
Objective O.CRYPTO_SECRECY ensures that any data related to the cryptographic functionality, such as keys, signatures, and algorithms is kept secret. Hereby, directives against loss, theft, or modification of data are provided.

#### B.8.1.2 Rationale to Uphold Assumptions

Following rationale aims at demonstrating how the objectives uphold the assumptions.

**A.CORRECT_DEVELOPMENT**
The way to uphold this assumption is by enforcing vulnerability analysis and tests upon the TOE.

O.VULNERABILITY_ANALYSIS is included in order to verify that design, implementation, and test of the TOE do not contain flaws. These vulnerability analysis should be enforced upon the TOE on regular basis during the phases of design, implementation and test. Furthermore O.VULNERABILITY_ANALYSIS ensures that the TOE has been analysed for vulnerabilities and that any vulnerabilities found have been removed or otherwise mitigated during the phases.

The objective O.SELF_TEST ensures that the TOE provides self-testing functionality for all TOE security functions (scans and tests) so that flaws and intrusions can be detected. This is done regularly when the TOE is operational.

**A.EXTERNAL_PARTY**
This assumption stated in the security environment, enables the TOE to trust other external parties and products. When applying products developed by external parties it is determined that the products must as minimum live up to the security requirements of the TOE itself. This means objectives for unique identification and authentication, data integrity, cryptographic functions being FIPS validated, and any cryptographic keys

being kept secure address this assumption and enforces the external parties and products to meet these objectives too, ie. the O.UNIQUE_IA, O.DATA_INTEGRITY, O.CRYPTO_FUNCTIONS, and O.CRYPTO_SECRECY ensure this assumption.
Proper installations and configurations are the job of administrators of the TOE. So OE.TRAIN also addresses the assumption in order to verify that administrators are reliable to do their work properly.
O.SELF_TEST also has an impact in realising the assumption, since a self test could detect flaws in the products and enforces appropriate measures to be taken.

**A.NO_EVIL**

The objective OE.TRAIN upholds this assumption. This means that administrators are well trained and competent through the OE.TRAIN objective. Since training can not be seen as part of the functionality of the TOE, and can not be directly evaluated, the objective is stated as an objective for the environment. This is a realistic choice as training procedures cannot be carried out by the TSF but rather the overall responsible for the TOE. Furthermore, in order to obtain a role it is implied that the individual is trustworthy.

**A.PHYSICAL**

Physical protection of the TOE is divided into 2 categories: *back-up of data* and *isolation of sensitive parts.*

The O.BACK-UP objective addresses the issue of data back-up. The objective states that the TSF will provide procedures for back-up of data of the TOE. The data trail especially must be recoverable at any time.

Protection and isolation of sensitive parts is addressed in the objective OE.ISOLATION. Notice again that the objective is stated as an objective for the environment. This is reasonable because of the fact that isolation cannot be done by the TSF but is something that the overall responsible for the TOE shall ensure. The objective states that isolation of physical parts of the TOE will be provided, such that the TOE is safe guarded against physical damages, intrusion, theft, etc.

### B.8.1.3 Rationale to meet Policies

Following rationale demonstrates how the objectives meet the OSPs.

**P.AUTHORISED_USERS**

O.UNIQUE_IA meets this OSP since it ensures that only authorised entities are allowed access to the TOE.

The objective O.DATA_INTEGRITY ensures that unauthorised modification, theft, or deletion of TOE data shall be prevented. Therefore this objective contributes in meeting this policy.

O.ACCOUNTABILITY_AND_AUDIT also covers an aspect of the policy, since it takes into account that if any unauthorised access is detected in audit records, administrators will be notified so that they can fix the problem and cover up the security hole. If not a proactive approach, then at least it gives some contribution in meeting the policy.

Which rights and privileges users are assigned according to the role they obtain, must be maintained by management procedures concerning roles. Furthermore, new people get employed and some get fired, so constant updates must be managed. Thus, the objective O.ROLE_MANAGEMENT has an influence on the P.AUTHORISED_USERS policy.

**P.USER_PRIVILEGES**

User privilege policies are met by the objective O.ROLE_MANAGEMENT. It is the only objective that concerns the management of user roles and privileges. It ensures that the TSF can provide a mechanism to control rights and privileges according to user roles, as identified in section A.2.5.

**P.ACCOUNTABILITY**

The objectives that address this policy are: O.UNIQUE_IA, O.ACCOUNTABILITY_AND_AUDIT and O.ROLE_MANAGEMENT.

The O.UNIQUE_IA objective has an influence in ensuring this policy, since obtaining accountability requires that entities are identified in order to be able to relate them to their activity and actions done within the TOE.

Generation of audit records in the TOE is ensured by the O.ACCOUNTABILITY_AND_AUDIT objective. The audit events are associated with the identity of users and their rights and privileges. Therefore accountability and audit is closely related to the objective O.ROLE_MANAGEMENT. Thereby, both O.ACCOUNT-ABILITY_AND_AUDIT and O.ROLE_

MANAGEMENT have an impact on the policy.

**P.CRYPTOGRAPHY**

To meet this policy there are two objectives, O.CRYPTO_FUNCTIONS and O.CRYPTO_SECRECY, that each concern two different aspects of the policy.

The O.CRYPTO_FUNCTIONS addresses the issue of implementing the cryptographic algorithms and functions using FIPS validated cryptographic services. It enforces the TOE data to be encrypted following some standard cryptographic services. The services provide confidentiality and integrity protection of TSF data as well as in transit (connections) and in end points (e.g. windmills and servers).

O.CRYPTO_SECRECY covers a different aspect of the policy. The objective states that key data or other executable code associated with the cryptographic functionality shall be kept secret. This includes keys, signatures, algorithms, etc.

O.SELF_TEST provides a partial upkeep to the policy since a self test can, when everything else fails, detect a flaw/error in cryptographic related code and alarm some suspicious activity and appropriate measures can be taken in order to safe guard against threats.

**P.TRAIN**

The only objective addressing this policy is the environment objective OE.TRAIN. Training has nothing to do with the functional operation of the TOE, and the OE.TRAIN is therefore an objective that must be satisfied by the operational environment of the TOE. The objective ensures that authorised users that hold a role within the TOE have received proper training and thus are found competent and trustworthy to operate the TOE in a secure way. Furthermore, users will get continuous training when new functions are incorporated in the TOE.

### B.8.2 Security Requirements Rationale

This section states the rationale behind the IT security functional requirements and security assurance requirements.

### B.8.2.1 Security Functional Requirements Rationale

The rationale for why the security objectives are met by specified SFRs is explained below, and an overview is shown in table B.5. Furthermore, table B.5 reasons why stated requirements are sufficient in order to meet all objectives and that each objective is covered by at least one requirement.

**O.UNIQUE_IA**

The FIA_UAU.2, FIA_UAU.3, and FIA_UID.2 components ensure that a user is identified and authenticated before being allowed any actions and thus identification and authentication of all entities are enforced by these components. FIA_UAU.6 ensures re-authentication when a user wants to resume a locked session.

The management of identification and authentication security functions and attributes are ensured by the FMT class (security management) from where the FMT_MOF.1, FMT_MSA.1, FMT_MTD.1, FMT_SMF.1, and FMT_SMR.1 components are selected.

The FMT_MOF.1 component specifies security functions and which users have the privileges to manage these [5].

Management of security attributes regarding identification and authentication in the SF is enforced by FMT_MSA.1.

FMT_MTD.1 takes care of management of TSF data, hereunder data concerning identification and authentication. This component restricts the ability to access the TSF data to only being available for authorised users that own special rights to read, modify, or delete TSF data.

Besides other management functions which the FMT_SMF.1 component specifies, the management function of identification and authentication is also included in this component.

**O.DATA_INTEGRITY**

Integrity of data in the TOE (user data and TSF data) is ensured by the FDP (user data protection) and FPT (protection of the TSF) classes.

With the FDP_IFC.1 component an information flow control SFP is specified. Additionally, this component defines the list of subjects (e.g. users, machines, or processes), information (e.g. email or network protocols), and a subset of the possible operations in the TOE that this policy shall be enforced upon.

| | O.UNIQUE_IA | O.DATA_INTEGRITY | O.ACCOUNTABILITY_AND_AUDIT | O.CRYPTO_FUNCTIONS | O.ROLE_MANAGEMENT | O.SESSION | O.BACK-UP | O.VULNERABILITY_ANALYSIS | O.SELF_TEST | O.CRYPTO_SECRECY |
|---|---|---|---|---|---|---|---|---|---|---|
| FAU_ARP.1 | | | × | | | | | | | |
| FAU_GEN.1 | | | × | | | | | | | |
| FAU_GEN.2 | | | × | | | | | | | |
| FAU_SAA.1 | | | × | | | | | | | |
| FAU_SAR.1 | | | × | | | | | | | |
| FAU_SAR.2 | | | × | | | | | | | |
| FAU_STG.1 | | | × | | | | | | | |
| FCS_CKM.1 | | | | × | | | | | | × |
| FCS_CKM.2 | | | | × | | | | | | × |
| FCS_CKM.4 | | | | × | | | | | | × |
| FCS_COP.1 | | | | × | | | | | | × |
| FDP_IFC.1 | | × | | | | | | | | |
| FDP_IFF.1 | | × | | | | | | | | |
| FDP_ITT.1 | | × | | | | | | | | |
| FDP_SDI.1 | | × | | | | | | | | |
| FIA_UAU.2 | × | | | | | × | | | | |
| FIA_UAU.3 | × | | | | | | | | | |
| FIA_UAU.6 | × | | | | | × | | | | |
| FIA_UID.2 | × | | | | | | | | | |
| FMT_MOF.1 | × | × | × | × | × | × | × | × | × | × |
| FMT_MSA.1 | × | × | × | × | × | × | × | × | × | × |
| FMT_MSA.2 | | | | × | | | | | | × |
| FMT_MSA.3 | | × | | | | | | | | |
| FMT_MTD.1 | × | × | × | × | × | × | × | × | × | × |
| FMT_SMF.1 | × | × | × | × | × | × | × | × | × | × |
| FMT_SMR.1 | × | × | × | × | × | × | × | × | × | × |
| FPT_AMT.1 | | × | | | | | | | | |
| FPT_ITT.1 | | × | | | | | | | | |
| FPT_RCV.2 | | | | | | | × | | | |
| FPT_STM.1 | | | × | | | | | | | |
| FPT_TST.1 | | × | | | | | | | × | |
| FTA_SSL.1 | | | | | | × | | | | |
| FTA_SSL.2 | | | | | | × | | | | |

Table B.5: Rationale for requirements satisfying security objectives.

FDP_IFF.1 enforces the information flow control SFP specified in FDP_IFC.1 based on types of subject and information security attributes. The list of subjects and information controlled under the indicated SFP, and for each of these their security attributes are specified in this SFR.

The component FDP_ITT.1 ensures protection of user data when it is transferred within the TOE, ie. via internal channels. Since the TOE in mind is a distributed system and is composed of various physically-separated parts (see section B.2) this component covers this kind of user data protection.

In order to maintain the integrity of stored user data in the TOE, FDP_SDI.1 is selected. It ensures protection of stored data by monitoring user data stored in containers controlled by the TSF for specified integrity errors on all objects, based on defined user data attributes.

The underlying abstract machine is a virtual or physical machine upon which the TSF executes. In order to verify the security assumptions, such as memory capacity and correct mode of operation, made about the underlying abstract machine the FPT_AMT.1 component specifies the conditions under which the verification has to occur by the TSF.

FPT_ITT.1 ensures integrity of TSF data that is being transferred between physically-separated parts of the TOE via internal channels.

In order to meet this objective, management issues have to be considered too especially regarding security attributes. Therefore, the components FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_SMF.1, and FMT_SMR.1 are included in the list of SFRs enforcing the O.DATA_INTEGRITY objective.

**O.ACCOUNTABILITY_AND_AUDIT**

This objective is mainly enforced by the FAU (security audit) class. From this class the component FAU_ARP.1 is selected, because when violation of TOE security has been detected, the TSF has to take some actions in order to correct the security violation. Responsible persons shall be informed. This component is dependant on the inclusion of the FAU_SAA.1 SFR.

In order to have a security alarm functionality in the TSF it is necessary that the TSF knows what to react upon in case of violation of security. With the FAU_SSA.1 component a set of rules in monitoring the audited events is specified which then will be used by the TSF to indicate a po-

tential violation.

The FAU_GEN.1 component of the security audit data generation (FAU_GEN) family defines requirements for the level of auditable events, and specifies the list of data that shall be recorded in each record [5].

FAU_GEN.2 is another component of the FAU_GEN family that ensures that the TSF is able to associate each auditable event with the identity of the entity that caused the event [5].
Since audit records are evidence of what has been going on in the TOE and can be used to detect unwanted activity it shall be possible for administrators to read audit information from the audit records, thus the FAU_SAR.1 component is included.
In accordance with the O.ACCOUNTABILITY_AND_AUDIT objective responsible users shall have access to view audit records. No one is allowed to modify or delete audit records since the audit data is significant for the maintenance of the TSF security.

FAU_STG.1 ensures protection of audit records from unauthorised access, modification, and/or deletion.

Furthermore, in order to ensure reliable time stamps for auditing and security attribute expiration, the FPT_STM.1 is needed.

The same management issues as in previous mentioned objectives are taken care of by the FMT class for the same reasons.

**O.CRYPTO_FUNCTIONS and O.CRYPTO_SECRECY**

When ensuring these objectives, it is not possible to look at the objectives separately since they are related to each other, ie. a SFR ensuring one objective depends on the presence of another SFR that satisfies the other objective.
The CC provides the cryptographic support class (FCS) for ensuring cryptography measures in the TOE.

FCS_COP.1 addresses the O.CRYPTO_FUNCTIONS objective. Through this SFR all cryptographic operations are required to be performed in accordance with a specified algorithm and with a cryptographic key of specified size.
By specifying of the use of cryptographic keys, FCS_COP.1 depends highly on how keys are managed in the TOE. These dependencies (FCS_CKM.1 and FCS_CKM.4) are provided for when ensuring the O.CRYPTO_SECRECY objective.
Furthemore, FCS_COP.1 is also dependent of the presence of secure secu-

rity attributes (FMT_MSA.2) because clearly cryptographic data such as keys are security attributes in them selves and thus only secure values are valid for these keys.

FCS_CKM.1 ensures that cryptographic keys are generated in accordance with a specified algorithm and key size. FCS_CKM.1 partly contributes to ensure objective O.CRYPTO_FUNCTIONS since it is not possible to ensure that cryptographic operations are performed properly if not cryptographic keys are appropriately generated.
Furthermore, FCS_CKM.1 also ensures O.CRYPTO_SECRECY since generation of keys must not be possible to be done by other parties. Any forgery of keys must not be possible and keys have to remain secret.

FCS_CKM.2 makes sure that cryptographic keys are distributed following a specified key distribution method. This is required since their exists several entities in the TOE model (ie. user/web clients, servers and windmills) that need to exchange data encrypted and mutually authenticate them selves to each other, and can only do so by exchanging cryptographic keys.

FCS_CKM.4 takes the destruction of cryptographic keys into account. It ensures that keys are destroyed appropriately with a clear destruction method. Just like generation, destruction contributes in ensuring both the objectives O.CRYPTO_FUNCTIONS and O.CRYPTO_SECRECY.

**O.ROLE_MANAGEMENT**

This objective is enforced by the security management class (FMT). The components included are FMT_MOF.1, FMT_MSA.1, FMT_MTD.1, FMT_SMF.1, and FMT_SMR.1.
The roles that are given to users have to be maintained by the TSF in order to keep the TSF secure. This is ensured by the FMT_SMR.1 component.

**O.SESSION**

Partly ensured by management requirements, the objective O.SESSION is also met by the SFRs listed below. The chosen components are needed in order to incorporate automatically and manually session locking. In order to manage security attributes of session locking some of the already described management SFRs are needed for the O.SESSION objective. These SFRs are the FMT_MOF.1, FMT_MSA.1, FMT_MTD.1, FMT_SMF.1, and FMT_SMR.1.

***FTA_SSL.1 - TSF-initiated session locking***

This component defines a session locking procedure carried out by the system. The period of time, that a session is allowed to be interactive before session locking, is also specified hereunder. The component specifies

furthermore the events that should occur prior to unlocking of the session.

***FTA_SSL.2 - User-initiated locking***
This component defines a session locking procedure carried out by the user. The component specifies furthermore the events that should occur prior to unlocking of the session.

Both FTA_SSL.1 and FTA_SSL.2 have dependency to FIA_UAU.1 (Timing of authentication). Furthermore the FIA_UAU.6 (Re-authenticating) SFR is necessary in order to cover O.SESSION.

**O.BACK-UP**
Components found to fulfill the O.BACK-UP security objective are the FMT_SMF.1 and FPT_RCV.2 components.
Among several other management functions, FMT_SMF.1 provides the means for an administrator to ensure continued operation of the TOE, including back-up and recovery. Thereby it specifies management functions for creating and recovering back-ups.

In the TOE considered in this project it is the data trail that has to be recoverable at any time. Back-up of data trail should be done on a regular basis since loss of data could mean costly damages. Therefore an automated approach is to be preferred.

FPT_RCV.2 specifies a list of failures/service discontinuities that the TSF shall recognise and react upon automatically.
This SFR is dependant on the inclusion of the assurance requirement AGD_OPE.1 which provides operational user guidance for back-up procedures too.

**O.VULNERABILITY_ANALYSIS**
This objective is ensured for by the assurance requirement AVA_VAN.2. It is clear that this objective can not be enforced by any functional requirement (excluding management issues that are discussed in the above mentioned objectives). AVA_VAN.2 ensures that vulnerability analysis are carried out on a regular basis to ascertain the presence of potential vulnerabilities.

**O.SELF_TEST**
In order to preserve the security of the TOE, the TSF has to periodically test its functionality and analyse whether it still is secure or not. This includes both detection of unauthorised entities (e.g. worms, vira, and spyware) and detection of flaws and errors in the various parts of the TSF (e.g. servers, file systems, and sensors). The FPT (protection of the TSF)

class ensures the O.SELF_TEST objective best since it focuses on protection of TSF data.

The FPT_TST.1 (TSF self test) component specifies conditions under which self test should occur and the integrity of which parts of the TSF should be verified. Thus the integrity and assurance of correct operation of TSF is preserved by this SFR. This component is dependant on FPT_AMT.1.

#### B.8.2.2 Security Assurance Requirements Rationale

This section states the rationale behind the security assurance requirements. The security assurance components and requirements are derived directly from the EAL. The chosen EAL level is based on following reasoning:

Firstly, the two lowest assurance levels (EAL1 and EAL2) only reflect basic assurance. Secondly, to have a product at level 5 or higher (EAL5 - EAL7) it is needed to rely upon underlying systems, among other the operating system. This means that they also must have at least same assurance as the TOE itself. Since it is not within the scope of this project to analyse these underlying systems, these EALs are not considered for this TOE. This only leaves EAL3 or EAL4 to be considered.
When comparing the two levels, it is important to take into consideration what the purpose of the TOE is, and under which circumstances and environment it will be deployed (see section B.2).
Furthermore, when looking closer at the assurance components that are different at the two levels, it is noticed that stronger demands during development, especially tests and vulnerability analysis which are identified as profound security objectives and a vital part of the environment in which the TOE is deployed, are in greater focus at EAL4 (e.g. in the AVA_VAN family).
During development more assurance is given by EAL4 than EAL3 by requiring a design description, an implementation specification/representation, and improved mechanisms/procedures that provide confidence that the TOE will not be tampered with during development or delivery. Especially an outline for an implementation representation is exactly what is aimed for in this project.
Assessing the context in which the TOE is to operate further indicates the choice of EAL. The TOE is to operate within a rather closed environment by predefined known users/roles, ie. people with windmill knowledge and who are company authorised. Violation of security could have severe consequences financially and physically, and can affect the individual living being, because the windmills contribute electricity to the power system and their operation is important to the

overall power supply[63]. So the TOE must ensure that windmills are functioning correctly. This means high assurance to its security is highly relevant.
Having said that, EAL4 would be the most likely choice, but there are some requirements defined in EAL4 which are beyond the scope of this project. This includes for instance considerations on how the TOE shall be delivered and demands for giving a subset of the actual implementation.
Therefore it is concluded that the level of assurance stated by EAL3 without any augmentation is found most appropriate and therefore chosen. Since a partly implementation representation of the TSF is aimed for in this project, the assurance requirement ADV_IMP.1 of EAL4 would have been ideal to include and thereby augment EAL3 with this component. But because ADV_IMP.1 has dependency on other components of EAL4, this is abstained from.
The included assurance components are predefined by the CC, and listed in table B.3.
Notice that the assurance component AGD_OPE.1 covers all objectives. This is due to the given definition in [7] section 13.1 which states that this component is an operational user guidance document. It describes the security functionality provided by the TSF and gives instructions and guidelines, and helps to understand the TSF. Furthermore, it includes the security-critical information and actions required for its secure use.

### B.8.3 TOE Summary Specification Rationale

The TOE summary specification rationale outlines the rationale for the security functions, the security functional policies, and the assurance measures of the TOE. These are described in the following sections.

#### B.8.3.1 TOE Security Functions Rationale

Table B.6 provides a mapping of security functions to SFRs for the TOE and is followed by a discussion of how each security function satisfies corresponding SFRs.

**F.BACK-UP**
This security function concerns back-up of data within the TOE. Thus, it meets the FPT_RCV.2 requirement.

[63]Much like the power circuit breakdown stated in [22].

| | F.BACK-UP | F.ROLE | F.AUDIT | F.AUTH | F.CRYPTOGRAPHY | F.SCAN | F.MANAGEMENT |
|---|---|---|---|---|---|---|---|
| FAU_ARP.1 | | | × | | | | |
| FAU_GEN.1 | | | × | | | | |
| FAU_GEN.2 | | | × | | | | |
| FAU_SAA.1 | | | × | | | | |
| FAU_SAR.1 | | | × | | | | |
| FAU_SAR.2 | | | × | | | | |
| FAU_STG.1 | | | × | | | | |
| FCS_CKM.1 | | | | | × | | |
| FCS_CKM.2 | | | | | × | | |
| FCS_CKM.4 | | | | | × | | |
| FCS_COP.1 | | | | | × | | |
| FDP_IFC.1 | | | | × | × | | |
| FDP_IFF.1 | | | | × | × | | |
| FDP_ITT.1 | | | | × | × | | |
| FDP_SDI.1 | | | × | | | × | |
| FIA_UAU.2 | | | | × | | | |
| FIA_UAU.3 | | | | × | | | |
| FIA_UAU.6 | | | | × | | | |
| FIA_UID.2 | | | | × | | | |
| FMT_MOF.1 | | | | | | | × |
| FMT_MSA.1 | | | | | | | × |
| FMT_MSA.2 | | | | | | | × |
| FMT_MSA.3 | | | | | | | × |
| FMT_MTD.1 | | | | | | | × |
| FMT_SMF.1 | | | | | | | × |
| FMT_SMR.1 | | × | | | | | |
| FPT_AMT.1 | | | | | | × | |
| FPT_ITT.1 | | | | × | × | | |
| FPT_RCV.2 | × | | | | | | |
| FPT_STM.1 | | | × | | | | |
| FPT_TST.1 | | | | | | × | |
| FTA_SSL.1 | | | | × | | | |
| FTA_SSL.2 | | | | × | | | |

Table B.6: Mapping of security functions to security functional requirements.

**F.ROLE**
This security function addresses management of roles in the TOE. Therefore, F.ROLE satisfies the FMT_SMR.1 requirement.

**F.AUDIT**
Besides covering the security audit functionality (ie. the selected SFRs from the FAU familiy) in the TOE, this security function satisfies also the requirements concerning reliable timestamps (ie. the FPT_STM.1 SFR) since audit uses time and date functionalities in order to register auditable events in the TOE.

**F.AUTH**
Since this security function addresses identification and authentication of entities the identified SFRs from the FIA family are satisfied by this security function. Session locking functionalities (ie. FTA_SSL.1 and FTA_SSL.2) are also met by F.AUTH since session locking requires reauthentication when unlocking a session. Furthermore, any authentication requirements within the DATA_FLOW_SFP is satisfied by this security function, ie. F.AUTH meets the requirements stated in FDP_IFC.1, FDP_IFF.1, FDP_ITT.1, and FPT_ITT.1.

**F.CRYPTOGRAPHY**
This component concerns cryptographic operations in the TOE. Therefore, the FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, and FCS_COP.1 are met by this security function. Additionally the FDP_IFC.1, FDP_IFF.1, FDP_ITT.1, and FPT_ITT.1 are satisfied by this security function since these SFRs cover data integrity and confidentiality within data flows as stated in the DATA_FLOW_SFP.

**F.SCAN**
Selftesting and scanning of the TOE data are covered by this security function. Thus, F.SCAN satisfies the FDP_SDI.1, FPT_AMT.1, and FPT_TST.1 SFRs.

**F.MANAGEMENT**
This security function meets management issues in the TOE besides the management of roles which is met by F.ROLE. Thus, this security function satisfies the FMT_MOF.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1, and FMT_SMF.1.

#### B.8.3.2 TOE Assurance Measures Rationale

The mapping of assurance measures to security assurance requirements is shown in table B.7.

| | M.ARCH | M.SPEC | M.DOCS | M.SETUP | M.AUTH_CTRL | M.ID | M.DELIVER | M.SECMEASURES | M.LIFE-CYCLE | M.CLAIMS | M.EXCOMP | M.INTRO | M.OBJ | M.REQ | M.SPD | M.TSS | M.TEST |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADV_ARC.1 | × | | | | | | | | | | | | | | | | |
| ADV_FSP.3 | | × | | | | | | | | | | | | | | | |
| ADV_TDS.2 | × | × | | | | | | | | | | | | | | | |
| AGD_OPE.1 | | | × | | | | | | | | | | | | | | |
| AGD_PRE.1 | | | | × | | | | | | | | | | | | | |
| ALC_CMC.3 | | | | | × | × | | | | | | | | | | | |
| ALC_CMS.3 | | | | | × | | | | | | | | | | | | |
| ALC_DEL.1 | | | | | | | × | | | | | | | | | | |
| ALC_DVS.1 | | | | | | | | × | | | | | | | | | |
| ALC_LCD.1 | | | | | | | | | × | | | | | | | | |
| ASE_CCL.1 | | | | | | | | | | × | | | | | | | |
| ASE_ECD.1 | | | | | | | | | | | × | | | | | | |
| ASE_INT.1 | | | | | | | | | | | | × | | | | | |
| ASE_OBJ.2 | | | | | | | | | | | | | × | | | | |
| ASE_REQ.2 | | | | | | | | | | | | | | × | | | |
| ASE_SPD.1 | | | | | | | | | | | | | | | × | | |
| ASE_TSS.1 | | | | | | | | | | | | | | | | × | |
| ATE_COV.2 | | | | | | | | | | | | | | | | | × |
| ATE_DPT.1 | | | | | | | | | | | | | | | | | × |
| ATE_FUN.1 | | | | | | | | | | | | | | | | | × |
| ATE_IND.2 | | | | | | | | | | | | | | | | | × |
| AVA_VAN.2 | | | | | | | | | | | | | | | | | × |

Table B.7: Mapping of assurance measures to security assurance requirements.

### B.8.4 Conformance Claims Rationale

This ST claims conformance to the CC part 2 and part 3 version 3.1 since functional and assurance requirements are taken directly from there. Furthermore, this ST claims conformance to the WPP (Windmill DMC System CC Protection Profile) since additions and enhancements relative to the PP have been specified and they do not reduce the PP security requirements.

# Bibliography


[1] *Common Criteria - An Introduction*. http://commoncriteriaportal.org.

[2] *Common Criteria for Information Technology Security Evaluation - Part 1: Introduction and general model*. CCMB-2005-07-001. Version 3.0 Revision 2.

[3] *Common Criteria for Information Technology Security Evaluation - Part 1: Introduction and general model*. CCMB-2006-09-001. Version 3.1 Revision 1.

[4] *Common Criteria for Information Technology Security Evaluation - Part 2: Security functional components*. CCMB-2005-07-002. Version 3.0 Revision 2.

[5] *Common Criteria for Information Technology Security Evaluation - Part 2: Security functional components*. CCMB-2006-09-002. Version 3.1 Revision 1.

[6] *Common Criteria for Information Technology Security Evaluation - Part 3: Security assurance components*. CCMB-2005-07-003. Version 3.0 Revision 2.

[7] *Common Criteria for Information Technology Security Evaluation - Part 3: Security assurance components*. CCMB-2006-09-003. Version 3.1 Revision 1.

[8] *http://www.fortinet.com*.

[9] *http://www.mozilla.com/en-US/firefox/*.

[10] *http://www.vestas.com.*

[11] *http://www.wikipedia.org.*

[12] *Open Source Tripwire®.* http://sourceforge.net/projects/tripwire/.

[13] *RSA Security Inc.* http://www.rsa.com/.

[14] *US Government Protection Profile - Wireless Local Area Network (WLAN) Client for Basic Robustness Environments.* March 2006 - Version 1.0.

[15] *Designing a Secure Point-of-Sale System - Assurance Argument*, November 2005.

[16] Khaled Alghathbar, Csilla Farkas, and Duminda Wijesekera. Securing uml information flow using flowuml. *Journal of Research and Practice in Information Technology*, 38(1), February 2006.

[17] IBM / Red Hat / atsec. Red hat enterprise linux 3 update 2 security target for capp compliance. Technical report, IBM / Red Hat / atsec, 2004.

[18] John Barkley. *Application Engineering in Health Care.* http://hissa.nist.gov/rbac/proj/paper/paper.html, May 1995.

[19] CESG. *Directory of INFOSEC Assured Products*, October 2006. http://www.cesg.gov.uk/site/publications/media/directory.pdf.

[20] P. Chown. *RFC 3268 - Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS).* Network Working Group, June 2002. http://www.faqs.org/rfcs/rfc3268.html.

[21] CISCO. *Understanding Simple Network Management Protocol (SNMP) Traps - Document ID: 7244.*

[22] Frances Cleveland. *IEC TC57 Security Standards for the Power System's Information Infrastracture - Beyond Simple Encryption.* Utility Consulting International (UCI).

[23] Bobby Crouch. Cryptography chip handles ssl traffic. *Network World*, July 2002. http://www.networkworld.com/news/tech/2002/0729tech.html.

[24] T. Dierks and C. Allen. *RFC 2246 - The TLS Protocol Version 1.0.* Network Working Group, January 1999. http://www.faqs.org/rfcs/rfc2246.html.

[25] Michael Horowitz. *LINUX vs. WINDOWS - A comparison of Linux and Windows.* http://www.michaelhorowitz.com/Linux.vs.Windows.html.

[26] Dan Capano DTS inc. Distributed control systems primer.

[27] Andrew C. Meyers. Jflow: Practical mostly-static information flow control. Technical report, Laboratory for Computer Science - Massachusetts Institute of Technology, January 1999.

[28] Arup Nanda. *Oracle Database 10g: Top Features for DBAs - Release 2 Features Addendum.* Oracle. http://www.oracle.com/technology/pub/articles/10gdba/index_r2.html.

[29] National Institute of Standards and Technology. *FIPS PUB 186-2 - Standards Publication 186-2 DIGITAL SIGNATURE STANDARD (DSS)*, January 27 2000. http://csrc.nist.gov/publications/fips/.

[30] National Institute of Standards and Technology. *FIPS PUB 140-2 - SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES*, May 25 2001.

[31] National Institute of Standards and Technology. *FIPS PUB 180-2 - Standards Publication 180-2 SECURE HASH STANDARD (SHS)*, August 1 2002. http://csrc.nist.gov/publications/fips/.

[32] Nicholas Petreley. Security report: Windows vs linux. Technical report, An independent assessment, 2004.

[33] W. Polk R. Housley, W. Ford and D. Solo. *RFC2459 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile.* Network Working Group, January 1999. http://rfc.net/rfc2459.html.

[34] Jan Stafford. Windows vs. linux/oss today, part 2: Linux experts see strong server, weak desktop. *SearchOpenSource.com*, 27 July 2005. http://searchopensource.techtarget.com/originalContent/0,289142,sid39_gci1110794,00.html.