Security in POS Systems

M. Sc. Thesis
30 ECTS-point
IMM-THESIS-2005-52

Supervisor: Robin Sharp
IMM, DTU

Allan Pedersen, s960710 Anders Hedegaard, s973492

1st August 2005

Abstract

When implementing a Point Of Sale (POS) system it has become increasingly common that the IT provider hosts the POS application on centralized servers not located at the point of sale. The access to the POS application is then provided via a client-server based system where the POS terminal (POS client) and the attached POS devices is continuously connected to the POS application server e.g. via the Internet. POS devices may include printers, bar code scanners, payment terminals, etc.

This thesis analyzes and defines the security requirements for such a system, using an approach based on the Common Criteria for Information Technology Security Evaluation (CC). A CC Protection Profile for a generalized POS system is developed. Furthermore, a CC Security Target for a secure interface between the POS application and payment terminal is developed. The Security Target claims conformance to the developed Protection Profile. Finally, a design example of the secure interface is described in order to show the applicability of the developed Security Target.

Keywords: Common Criteria, Protection Profile, Security Target, Security Evaluation, Point of Sale, POS system, Payment Terminal.

Preface

This thesis documents the M.Sc. project by Anders Hedegaard Olsen, s973492 and Allan G. H. Pedersen, s960710. The project has been carried out from 1st February to 1st August at Department of Informatics and Mathematical Modelling (IMM), Technical University of Denmark (DTU). The project has been supervised by professor Robin Sharp, IMM, DTU.

We would like to thank Robin Sharp for many hours of useful supervision and Christina Malling for heroic review of the thesis.

Lyngby, 1st August 2005

Anders Hedegaard Olsen

Allan G. H. Pedersen

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation

Point Of Sale (POS) systems for handling payments are widely deployed in commercial outlets of all types, from petrol stations and kiosks to department stores and marts. Typically, at the actual point of sale there are one or more PCs, to each of which is attached a cash register, printer(s), bar code scanner and a payment terminal for credit card transactions. The PCs typically operate as clients for a POS application server, either locally or at a remote site.

When the business application server is located at a remote site, additional considerations regarding data security are required. This is particularly the case when payment terminals are integrated in the POS system.

1.2 Problem Statement

The aim of this project is to analyze and define the security requirements of a POS system, using an approach based on the Common Criteria for Information Technology Security Evaluation (CC). This shall be done by realizing the following three objectives.

1. A Protection Profile (PP) shall be developed in order to define the security requirements for POS systems in general.
2. A Security Target (ST) for a secure interface between a payment terminal and the remote POS application server shall be developed. The ST shall comply with the security requirements defined in the POS systems PP.
3. A design example shall be developed on basis of the ST in order to demonstrate that a realistic set of security requirements and measures have been stated.

1.3 Organization

The organization of the thesis is based upon the three objectives defined in the problem statement. Individual chapters and appendices are introduced below.

Chapter 1 This chapter describes the motivation and problem statement. In addition general introductions to POS systems and the Common Criteria are presented.

Chapter 2 This chapter describes the development of the POS systems PP. First, the PP development approach is described and then the stated security requirements are walked through chronologically with respect to the developed PP.

Chapter 3 This chapter describes the development of the ST. First, the ST development approach is described and the the stated security requirements are walked through chronologically with respect to the developed ST.

Chapter 4 This chapter presents a design example on basis of the ST.

Chapter 5 This chapter states the overall conclusion, discussion, and perspective of the thesis.

Appendix A This appendix contains the developed PP.

Appendix B This appendix contains the developed ST.

1.4 Terminology

The terminology used throughout the thesis is based upon that of the Common Criteria (CC) as described in CC part 1 section 2 [CC104]. Furthermore, it is assumed that the reader posses elementary knowledge of IT systems and IT security.

1.4.1 Abbreviations

The following abbreviations are used throughout the report.

3DES Triple DES

AES Advanced Encryption Standard

CBC Cipher-Block-Chain

CC Common Criteria

CCITSE Common Criteria for Information Technology Security Evaluation

COM Component Object Model

CTCPEC Canadian Trusted Computer Product Evaluation Criteria

DES Data Encryption Standard

EAL Evaluation Assurance Level

EDE Encrypt-Decrypt-Encrypt

FIPS Federal Information Processing Standards

ICC Integrated Circuit Card

IT Information Technology

ITSEC Information Technology Security Evaluation Criteria

LAN Local Area Network

MAC Message Authentication Code

MSC Magnetic Stripe Card

NTP Network Time Protocol

OSP Organizational Security Policy

OTRS Open Terminal Requirements Specification

PA Payment Application

PAC Payment Application Client

PAN Primary Account Number

PED PIN entry device

PIN Personal Identification Number

POS Point Of Sale

POSPP Point Of Sale CC Protection Profile

PP Protection Profile

PSAM Purchase Secure Application Module

RSA Rivest, Shamir and Adleman

RSAENH The Microsoft Enhanced Cryptographic Provider

SFP Security Function Policy

SFR Security Function Requirement

SHA Secure Hash Algorithm

SHS Secure Hash Standard

SOF Strength Of Function

ST Security Target

TAPA Terminal Architecture for PSAM Applications

TCSEC Trusted Computing System Evaluation Criteria

TLS Transport Layer Security

TOE Target Of Evaluation

TSF TOE Security Function

TSP TOE Security Policy

1.5 POS systems

The point of sale is the physical location at which goods are sold to customers. Point of sale (POS) systems are systems designed to register sales and payments at the point of sale when the goods are sold. POS systems may be designed in many ways depending on the point of sale, which goods are sold, and whether the POS system is attended or unattended. A POS system may be large complex systems with many POS terminals working together, or it may be small single cash register systems.

1.5.1 Attended POS Systems

Attended POS systems are the most common and are used in almost any store and supermarket. In a supermarket the POS terminals usually are implemented as a line of check-out counters which are designed to operate effectively with a high flow of customers and goods. The operator of the POS terminal will usually not be able to serve the customers in other ways than handling the sales and payments.

In stores and kiosks with a higher level of customer service the POS terminals are typically implemented as more traditional cash registers and they may be distributed physically around the store. POS terminals may even be small hand-held devices e.g. for restaurants and ticket inspectors in trains.

1.5.2 Unattended POS Systems

Unattended POS systems are becoming still more widespread. In particular for selling tickets e.g. for movies, personal transportation, sports events, theme parks and zoos, but also self-service check-outs are beginning to find their way in some visionary supermarkets. Tickets are well suited goods for unattended POS systems because the POS terminal can produce the actual goods for sale on location when the payment have been validated. The systems for producing the goods and validating payments must, however, be very reliable. The POS terminal may also contain prefabricated goods and provide it upon payment validation but these systems have a limited area of suitable goods to handle, e.g. petrol.

More general purpose unattended POS systems may depend on reliable costumers to do the registering of goods correctly. Radio Frequency Identification (RFID) price tags may be a solution for this problem in the near future as both the tags and the receivers becomes more cost-effective.

1.5.3 POS Devices

Any POS system have miscellaneous POS devices attached. These may be more or less directly integrated with the POS terminal or cash register depending on hardware design. The attached POS devices usually includes keyboard, bar code scanner, payment terminal for debit/credit cards, customer and operator displays, and receipt printers. In case of an unattended POS system a coin and bank note counter may be necessary. Some devices are used for registration of the sales and payment and is called input devices. Other devices are used for providing evidence of the sales and payments registered and is called output devices.

1.5.4 Audit Trail

When the sales and payments are registered with the input devices the information is stored in the audit trail. The audit trail is the definitive evidence of the financial transactions performed within the POS system. The minimum quality and amount of information as well as how and for how long it is stored may be dictated by legislation. The information processed and stored in the audit trail may however be arbitrarily detailed and comprehensive depending on the needs and functionality of the POS system.

1.5.5 IT POS systems

One of the great advantages of IT based POS systems is the possibility to integrate it with standard business capabilities like financial accounting, stock and purchase management, etc. For this reason, most IT POS systems are fully or partially integrated with a standard financial accounting system. In this way the merchant is able to collect all the administrative tasks in one application without having to do the accounting of the POS audit trail manually.

A fully integrated POS application is implemented as an add-on application or module in the general financial accounting application. A partially integrated POS application is a stand-alone application with the ability to export the financial part of the audit trail, in order to register the financial postings automatically, by importing this into the financial accounting application.

1.5.6 Hosted IT POS Systems

When implementing a POS system it has become increasingly common that the IT provider hosts the POS application on centralized servers not located at the point of sale. The access to the POS application is then provided via a client-server based system where the POS terminal (POS client) is continuously connected to the POS application server e.g. via the Internet. This construction is particularly popular when the POS application is fully integrated with the financial accounting system and several benefits may be obtained from a hosted solution, e.g.:

Cost Most stores do not find it profitable to invest in complicated client-server based POS and accounting systems to handle typically one or two cash registers. With the hosted solution several stores have the option to share the costs of investment and maintenance of servers, storage, and software and hereby the possibility to invest in more robust and professional equipment opens.

Store chain Obvious benefits can be drawn from the hosted solution for store chains. The individual stores are then able to share data and the application software will be completely homogeneous from store to store. Owners and auditors can have direct access to all financial data for any store in the chain and synchronization, e.g. of gift vouchers and credit notes, is straightforward.

Administration The hosted hosted system makes the administration of the POS system is easier for the IT provider. Administrators are likely to be more skilled when the IT provider is maintaining the system, rather than if the individual store is responsible for security administration, back-up routines etc.

1.6 IT Security Engineering using the Common Criteria

When engineering secure IT systems an approach of modeling security requirements and evaluation requirements is needed. The commonly used approach today is that of the Common Criteria.

1.6.1 Background

As the use of IT systems increased during the 1970's an 1980's a need for security in the systems became clear. Therefore, the United States started to develop a criteria for evaluation of IT security in the early 1980's. This is known as Trusted Computer System Evaluation Criteria (TCSEC[1]). In the 1990's different countries began developing their own criteria based on the TCSEC, but more flexible and adapting to the evolving nature of IT in general.

In Europe several countries: France, Germany, the Netherlands, and the United Kingdom, joined their efforts to develop a criteria for IT evaluation. These countries had been developing their own criteria but they realized that if they worked together they could develop a criteria that would stand stronger. This resulted in the Information Technology Security Evaluation Criteria (ITSEC).

In Canada, the Canadian Trusted Computer Product Evaluation Criteria (CTCPEC) was developed. This criteria combined the TCSEC and ITSEC approaches.

Eventually, all attempts to develop a criteria led to a cooperation towards the production of the Common Criteria (CC). The CC version 1.0 was released in January 1996. Two years after in April 1998 version 2.0 was released.

The sponsoring organizations cooperated with the International Organisation for Standards[2] (ISO) in order to ensure that the CC is suitable to become a formal standard. As a result of this the CC version 2.1 is now recognized as ISO 15408. The fact that the CC is formalized to an ISO standard makes the use of the CC as the preferred IT security evaluation criteria spread quickly.

1.6.2 Target Audiences

In general there are three groups with an interest in IT security evaluation. These groups are *Consumers*, *Developers*, and *Evaluators* and they are described in the following.

Consumers The CC is developed to satisfy the needs of the consumers, as this is the fundamental purpose and justification for an evaluation process. Consumers may use evaluations to compare different products or systems. The CC

[1] Commonly known as the Orange Book.

[2] www.iso.org

gives consumers an implementation independent structure, named the Protection Profile (PP) in which to express their special requirements for IT security measures.

Developers Developers use the CC to identify security requirements to be satisfied by a product under development. The requirements for this product are contained in an implementation dependent document called the Security Target (ST).

Evaluators When evaluators need to evaluate a product the CC provides a set of general actions the evaluator needs to take.

1.6.3 Organization of the Common Criteria

The CC is divided into three parts which are listed below with a short description of each.

Part 1, Introduction and general model This part [CC104] introduces the CC. General concepts and principles are defined and a general model of evaluation is presented.

Part 2, Security functional requirements This part [CC204] defines a set of security functional components as a standard way of expressing the security requirements for IT products and systems. The catalogue is organized into classes, families, and components.

Part 3, Security assurance requirements This part [CC304] defines a set of assurance requirements which can be used as a standard express the assurance requirements for IT products and systems. As in part 2 the requirements are catalogued and organized into classes, families, and components. Furthermore, evaluation criteria for PPs and STs are defined and the evaluation assurance levels (EALs) are presented. These are predefined packages of assurance components that make up the scale for rating confidence in the product.

1.6.4 Protection Profile

According to the CC a Protection Profile is defined as follows.

> *An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs*[3].

A Protection Profile describes a set of security requirements which are supposed to counter specified threats to a Target Of Evaluation (TOE) in a specified environment.

A PP may be written by several parties. User communities may write a PP to

[3] [CC104] p. 14.

define the security requirements of a related group of TOEs, i.e. firewalls, operating systems, etc. A PP may also be written by a large organization such as a government, as a way of securing a minimum level of security for a similar group of TOEs.

As a PP is implementation independent, and thereby a general description of security requirements for a TOE, it is designed to be reused.

1.6.5 Security Target

According to the CC a Security Target is defined as follows.

> *A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE*[4].

A Security Target is a specification of security requirements for a given TOE. A ST is similar to a PP in terms of structure, but in addition it contains information about product specific details.

Security requirements in the ST may be stated by reference to a PP, directly by reference to CC functional and assurance components, or stated explicitly.

[4] [CC104] p. 15.

CHAPTER 2

Protection Profile

In this chapter the development of the Protection Profile (PP) is described. The PP is attached as appendix A and it is recommended to read this simultaneously as the sections of the PP are walked trough chronologically.

2.1 PP Development

This section describes how the PP development process have been approached.

2.1.1 Contents of a Protection Profile

The Common Criteria (CC) clearly defines the content of a PP and dictates the order in which the content must be presented. How this is to be carried out is presented in appendix B in the CC part 1 [CC104]. In the following this will be walked through.

A PP is divided into six main sections which are:

1. PP Introduction — This section contains formalities such as identification and overview of the PP.

2. TOE Description — This section contains a description of the Target Of Evaluation (TOE) in order to make the reader understand its security requirements. The description shall include the product type and general IT features of the TOE.

3. TOE Security Environment — This section presents the security aspects of the environment in which the TOE is to operate. This includes the threats against the TOE, assumptions regarding the intended use of the TOE, and organizational security policies (OSPs) to which the TOE must comply.

4. Security Objectives — This section describes security objectives stated to counter the threats and address the assumptions and OSPs stated in the TOE Security Environment section.

5. IT Security Requirements — This section states the TOE security functional requirements (SFRs) which are found to satisfy the security objectives. It also states the Evaluation Assurance Level (EAL) which defines the TOE security assurance requirements. These are the requirements assuring that the TOE does indeed comply with the IT security requirement.

6. Rationale — This section is used to prove that all threats are countered by one or more security objectives (security objectives rationale). It also specifies the security requirements rationale which is used to prove that all security objectives are satisfied by one or more SFRs (the SFR rationale).

As indicated above the structure of a PP is straightforward in the sense that it follows the natural development process. First, the TOE and the environment in which the TOE is to operate are described. This description forms the basis for the analysis used to determine threats, assumptions, and OSPs. After this, security objectives used to counter the identified threats and address the stated assumptions and OSPs are defined. The next thing to do is to choose SFRs from [CC204] in order to satisfy the security objectives. In addition, the assurance requirements used to assure the claimed EAL are matched are stated. In the end, a rationale for the security objectives and a rationale for the security requirements are made in order to prove and justify that all threats are countered and that all objectives are satisfied.

2.1.2 Literature

Not much literature can be found on the Common Criteria and how to develop Protection Profiles except that of the CC part 1-3 itself. Part 1 [CC104] provides an introduction to the CC as well as the above mentioned structure and detailed descriptions of each section in a PP. Part 2 [CC204] is the catalog containing all the predefined security functional requirements from which the PP developer may select the relevant components to be used in the PP. Part 3 [CC304] defines the EALs and contains the catalog describing the security assurance requirements they consist of.

The CC part 1-3 are however quite extensive and may seem difficult to approach. Not much help is found for the inexperienced PP developer to get started. But the Common Criteria website[1] includes a directory of already developed and evaluated PPs, which can be of inspiration.

No POS system PP has previously been developed. The only directly POS related PP to be found in the PP directory is the *APACS PIN Entry Device Protection Profile* [APA03] which describes the security requirements for the communication between the PIN entry device and the card reader in a payment terminal. This PP has been very helpful and the organization of it was used as the overall template because it was considered well organized and approachable.

Other PPs address TOEs with similar functionalities and those from which most inspiration was found are the *Discretionary Information Flow Control (MU)* Protection Profile [DSLV02] and the *Commercial Database Management System Protection Profile* [Ora98]. [DSLV02] gave inspiration when constructing requirements for the *Data Flow Control Policies.* [Ora98] addresses related data access IT functionalities and gave inspiration when the TOE security environment, in particular, was defined.

The following sections describe how the PP was developed. The organization of the sections corresponds to the PP contents as described in section 2.1.1.

[1] www.commoncriteriaportal.org

2.2 The Target of Evaluation

The goal is to develop a Common Criteria Protection Profile (PP) which any type of IT POS system can claim conformance to. To do this, the first step is to make a general model describing the common features of any kind of IT POS system as described in section 1.5. This generalized POS system model will then be the Target Of Evaluation (TOE) defined in section A.2 in the PP.

2.2.1 The POS System Model

The general purpose of any POS system can be generalized into two main operations, which are to:

1) Register sales and payments of goods in the audit trail.

2) Produce evidence of sales and payments from the audit trail.

How these operations are performed highly depends on the nature of the point of sale, which goods are sold, and whether the POS system is attended or unattended. The operations are performed by means of input and output devices which causes data to flow into or out of the audit trail. The data flows are defined as the *input/output device data flows*. The audit trail is the data storage containing all financial transactions performed on the POS system. In a (secure) IT POS system the audit trail will also contain the security audit records which may be generated.

This general model of an IT POS system is illustrated in figure 2.1. The POS application is the central software component implementing the IT functionality and security of a POS system. The left side illustrates the registration operation, which causes the input data flows, and the right side illustrates the evidence operation, which causes the output data flows.

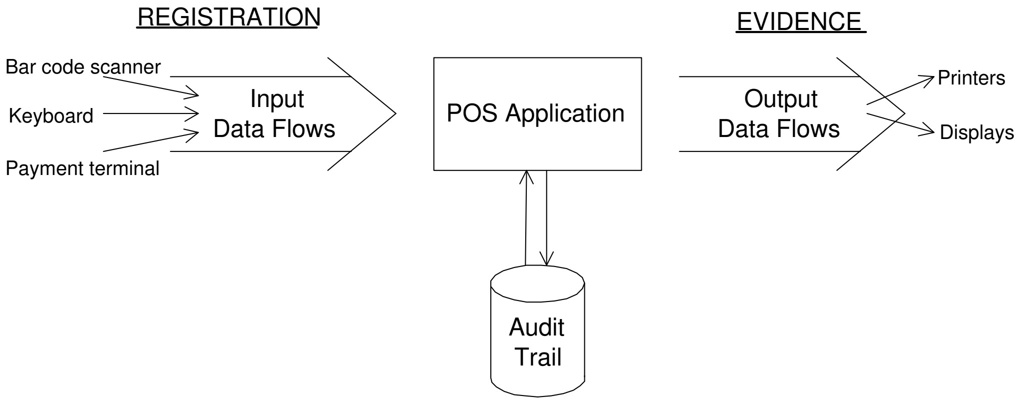

Figure 2.1: Generalized POS system model with input and output data flows.

2.2.2 POS System Roles

Different users interact with the POS system and even in a very generalized POS system they can be categorized into roles. The roles interact differently with the POS system and have different responsibilities to uphold the security of the system. The following roles are identified:

Customer The customer is a person who wants to purchase goods and does not, as such, have any responsibility in upholding the security of the POS system. The customer will usually only interact with the POS system indirectly via the operator, unless it is an unattended system where the customer also acts as the operator role (section 1.5.2. However, in some specialized operations the customer may need to interact directly with the POS system even if the system is attended by an operator. This could be when performing a payment card transaction on a payment terminal and PIN card-holder verification method is used.

Operator The operator is responsible for handling and registration of the sold goods and received payments, e.g. the sales clerk in a store or the waiter in a restaurant. Furthermore, the operator is responsible for providing evidence of the sales and payments to the customer.

Financial Manager The financial manager is the role authorized to extract the financial data from the audit trail to make financial accounting and balancing.

Administrator The administrator is overall responsible for upholding the security of the POS system. The administrator is authorized to install, configure and maintain any function in the system.

Other roles may be identified, e.g. by making a more detailed division of users. The mentioned roles are the minimum division in which the POS system should distinguish the users.

2.2.3 POS Devices and Data flows

Any IT POS system complying with the POS system model will have some attached input and output devices, which implements the user interfaces of the POS system. This could be keyboards, displays, bar code scanners, payment terminals, etc. Even if e.g. the keyboard is physically integrated into the POS terminal it is still categorized as a POS device.

When a device is used it creates a data flow between itself and the audit trail. The data will, however, normally not flow directly to and from the audit trail, as some data processing by the POS application usually is needed. This could be price lookup when an item is bar code scanned or print generation when producing a receipt for the customer.

Figure 2.2 illustrates how a POS device is interfaced with the POS application. Again, the model is generalized to be compatible with most implementations. In this model the device stub of the interface is a device driver of the POS device, but of course other implementations are possible depending on the general design of the POS system. Normally, each POS device will have its own POS application interface but related devices may share interfaces and device drivers. By way of example some PC based IT POS systems have bar code scanners and card readers implemented as keyboard extensions, hence it will be difficult to separate data flows originating from the keyboard and the extensions. If data flows from different devices must be separated to uphold the security policy, shared interfaces should be avoided.

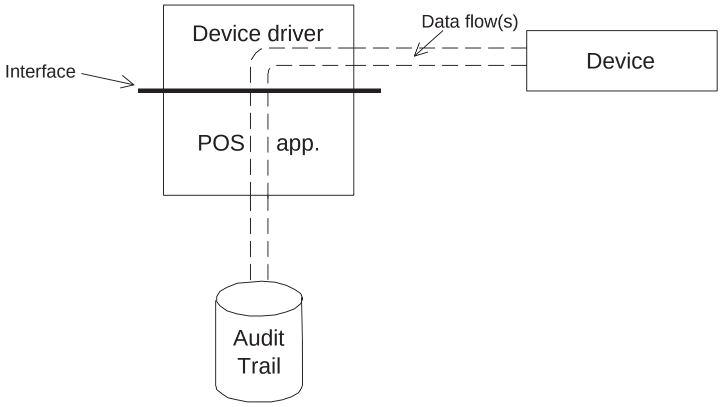

Figure 2.2: Interface between POS device and POS application.

The requirements for protecting the individual data flow may vary depending on several conditions. It is therefore necessary to perform a threat analysis for each identified data flow in order to determine the necessary level of protection needed to protect the data it contains. The threat analysis shall identify the probability and consequences of an attacker trying to compromise the security of the data flow. The PP defines the following attributes to be considered when performing the threat analysis:

Input/output device Some POS devices may be in greater risk than others of an attacker trying to compromise the security of these and the data they processes, i.e. the data flow. E.g. an attack against a payment terminal may be more likely than an attack against a customer display.

Roles The source and destination roles of the data should be considered as some roles may be more trusted than others.

Data The actual data or information flowing is very important to consider. The data may be more or less sensitive in relation to integrity and/or confidentiality.

Media The media in which the data is transmitted should also be considered in the analysis. For instance, some devices may be interfaced with the POS application via insecure LANs or wireless communication protocols from which security issues can arise.

Threat agents When performing a threat analysis it is always important to consider the potential threat agents and what they might want to obtain by compromising the security of the system.

Other attributes may also be relevant depending on the actual implementation, hence the list may be subject to additions during development of the POS system design.

When the threat analysis is carried out the level of protection of the data flow can be selected from the following list (definitions from [Whe05]):

Low level of protection states that minimum standard countermeasures are required to achieve desired security of the data flow.

Medium level of protection states that additional countermeasures above the minimum level of protection are required in order to achieve desired security.

High level of protection states that most stringent and rigorous security countermeasures are required.

It is not defined which countermeasures are categorized under low, medium and high level of protection respectively as this again depends on the nature of the actual POS system and the POS devices implemented. The countermeasures shall be defined in addition to the selection of level of protection. The countermeasures may include physical protection, use of cryptographic techniques, and mandatory operator actions. Altogether, the threat analysis, the level of protection, and the countermeasures required to uphold desired security of the data flow are defined as the *Input/Output Device Data Flow Control Policy.*

2.2.4 Special Conformance Claims

As described earlier it is a goal to develop a PP which any type of POS system should be able to comply with. As many POS systems are composed of several components, which may be developed by different providers, it will also be a goal to develop the PP in a way that enables these different vendors to claim conformance to the PP when only a part of the POS system is developed. This could be if a provider wants to develop a POS application and not definitively determine exactly which and how many POS devices to be attached in the final implementation. Or it could be a third party POS device vendor which may deliver its devices for many different POS application providers.

The CC states that a TOE cannot claim partially conformance any to PP or ST.

However, the CC offers the possibility to state requirements for the TOE environment and in particular the IT TOE environment. The partially conformance claims are then solved in the PP by opening the possibility to move the security requirements, which is not to be complied by the TOE in question, to its IT environment. This means that if the IT environment is complying with the requirements which is not covered by the TOE, PP compliance claims will be possible. The interface model illustrated in figure 2.2 may be used to distinguish between the IT environment and the TOE.

2.3 TOE Security Environment

This section describes the development of the TOE security environment of the PP. The TOE security environment shall describe the security aspects of the operating environment in which the TOE is to operate. All assumptions, assets, threat agents, threats, and organizational security policies (OSPs) stated in the PP are described as these are the foundation of the security objectives.

2.3.1 Assumptions

Assumptions are to be met by the TOE environment in order for the TOE to be considered secure.

All users interacting with an IT system are potential attackers. Therefore, there must be an assumption assuring that at least one user can manage and maintain the security of the functions and data it contains in a competent manner. In addition this person must be assumed not to have evil intentions. As administrators manage and maintain the IT system it comes naturally to make the following assumption:

A.NO_EVIL *It is assumed that administrators of the TOE are competent of managing and maintaining the TOE and the security of the functions and data it contains. It is also assumed that administrators do not have evil intentions of abusing their privileges.*

It is necessary to make this assumption for almost any TOE where installation and configuration are needed or where any security function is manageable.

2.3.2 Threats to Security

This section describes the assets to be protected by the TOE, the threats agents, and the threats against the assets.

2.3.2.1 Assets

The primary asset of the TOE is derived directly from the purpose of the TOE. A (generalized) IT POS system is defined as an IT system designed to do the following operations[2]:

1) Register sales and payments of goods in the audit trail.

2) Produce evidence of sales and payments from the audit trail.

As the POS system shall be able to produce evidence of the registered sales and payment from the audit trail, e.g. for the financial accounting, loss or malicious manipulation of this may lead to conflicts with legislation and thereby cause trouble

[2] Section 2.2.1

for the owner. Furthermore, information stored in the audit trail is valuable to the owner in terms of sales statistics and other financial information. This information may also be valuable for attackers in relation to industrial espionage.

Additionally, in order to uphold the security of the POS system, security attributes need to be protected from disclosure and manipulation, e.g user names and passwords, cryptographic keys, etc.

As the POS system revolves around the audit trail and the security attributes are "merely" used to uphold the security of the POS system, it can be concluded that the audit trail is the primary asset to protect and the security attributes are secondary, though no less important, assets to protect.

2.3.2.2 Threat Agents

To define the threats against a POS system it is necessary to identify the threat agents, i.e. individuals with an interest in compromising the security of a POS system.

Threat agents are divided into two groups; *authorized* and *unauthorized users*. Authorized users are typically individuals motivated by personal revenge or economic gain, e.g. if an employee gets fired there may be an urge for this individual to harm the employer. Unauthorized users may be the typical hacker or cracker with an interest in compromising the security for economic gain, espionage, or even fun. Both authorized and unauthorized threat agents are referred to as *attackers*.

2.3.2.3 Threats

This section describes how the threats stated in section A.3.2 of the PP are found. The text in *italic* is the threats as they are stated in the PP.

T.ACCESS *An attacker may try to gain unauthorized access to the information protected by the TOE. This could be an unauthorized user impersonating an authorized user, or it may be an authorized user impersonating a, perhaps, more privileged user.*

This threat appears as unauthorized access to the TOE poses to be one of the major threats against the security of the TOE. The access may be in form of a typical hacker attack where an unauthorized user finds a way through the security measures, thereby gaining access to restricted areas. Another type of unauthorized access may be an authorized user impersonating a user with, perhaps, more privileges. E.g. a person who has found or stolen a user name with an associated password. Authorized users gaining unauthorized access pose a threat as they may see information which they are not authorized to see.

T.MODIFICATION *An attacker may try to modify information protected by the TOE maliciously.*

As opposed to T.ACCESS this threat deals with the problem that an attacker actually tries to modify information protected by the TOE, and in particular the audit trail. If data is maliciously modified, e.g. if cryptographic functions are implemented and the attacker modifies the cryptographic keys, the security of the system is seriously compromised. If information contained in the audit trail is modified with evil intentions it is the foundation for the financial accounting which is being modified, causing incomplete financial accounting.

T.PHYSICAL *The audit trail may physically be lost due to fire, theft, force majeure, etc.*

As the POS system revolves around the audit trail it poses a threat if the audit trail is physically lost. If it is lost the POS system breaks down and becomes useless. This threat covers all cases where the audit trail is physically lost, e.g. fire and theft[3].

T.UNATTENDED_SESSION *An attacker may gain unauthorized access to the TOE via a unattended session.*

If an authorized user leaves a session without shutting it down it leaves the session open for an attacker to gain unauthorized access to the TOE. An unattended session may, for instance, occur in a department store where the sales clerk leaves the counter to help a customer finding a nice pair of pants. An attacker can then take advantage of the inattentive moment and the unattended session.

T.INCOMPETENCE *A user may compromise the security of the TOE due to incompetent usage.*

Incompetence poses a threat in the sense that a user may use the POS system in a way which is not intended, thereby compromising the security simply because the user does not know better. This threat is common during holidays if the permanent staff are replaced by temporary staff or other employees not trained for POS operation.

T.DATA_FLOW *An attacker may compromise the integrity of an input/output data flow.*

If an attacker compromises the integrity of the data flows it causes the same trouble as if the audit trail was compromised. If the data flowing into the audit trail has been altered on the way, e.g. the transaction amount approved by a

[3] As well as abduction by aliens.

payment terminal is modified, there will be errors in the financial accounting. The data flows may also be altered when flowing out of the audit trail, e.g. if the data flow is from the audit trail to a receipt printer. This means that the produced evidence to the customer is wrong.

2.3.3 Organizational Security Policies

The Organizational Security Policies (OSPs) states additional rules, procedures and guidelines to be countered by the security objectives. The following OSPs are found necessary for a secure POS system:

P.AUTHORIZED_ USERS *Only authorized users may access the TOE.*

This policy is made to ensure that only users which are authorized can access the functionality of the TOE. This includes authentication and identification of users. By this policy anonymous access to the TOE is prevented.

P.ACCOUNTABILITY *Authorized users of the TOE shall be held accountable for their actions within the TOE.*

This policy is made to ensure that administrators can see the actions which have been taken in the TOE and attach a user to these actions. In this way a user of the POS system can be held responsible for the actions done and if deliberate fraud is committed, actions can be taken.

P.TRAIN *Authorized users accessing functions of the TOE shall receive continuous training in secure use of the TOE.*

This policy assures that all authorized users of the TOE will be capable of operating the TOE securely.

2.4 Security Objectives

This section explains the security objectives stated in the PP to counter the threats and address the assumptions and OSPs defined in the TOE security environment. Every assumption, threat, and OSP shall by addressed be at least one security objective and vice versa. The following security objectives are defined:

O.IA *The TOE shall provide means for identifying and authenticating users before allowing access to the TOE and its resources.*

This security objective has been identified mainly to counter T.ACCESS and T.MODIFICATION. By assuring that users of the TOE are identified and authenticated before they can access the TOE, unauthorized access to the information protected by the TOE is prevented. Furthermore, as a user has to be authorized to modify protected information the assurance of identification and authentication guard against unauthorized modification.

In addition this objective addresses P.AUTHORIZED_USERS as this OSP states that users shall be authorized to access the TOE. P.ACCOUNTABILITY is indirectly addressed as the users can not be held accountable without being identified first.

O.MANAGE *The TOE shall provide functionality which enables authorized administrators to manage and support the security attributes of the TOE, and restrict these functions from unauthorized use.*

This security objective has been identified to assure that the TOE's security functions are manageable only by authorized administrators. It counters T.ACCESS, T.MODIFICATION, and T.DATA_FLOWS indirectly by ensuring that the TOE supports management of security attributes, e.g. for user names and passwords, role based access control, and protection of the data flows respectively.

Additionally, the OSPs P.ACCOUNTABILITY and P.AUTHORIZED_USERS are addresses for the same reasons.

O.AUDIT *The TOE shall provide functionality to record security relevant events in sufficient detail to help administrators of the TOE to hold individual users accountable for any actions they perform that are relevant to the security of the TOE.*

This objective is mainly identified to address P.ACCOUNTABILITY. It assures that functionality to log security relevant events is provided by the TOE. This makes it possible to hold users responsible for their actions within the TOE. The audit level of detail is left to the administrator to decide.

It indirectly counters T.ACCESS and T.MODIFICATION because if attackers successfully compromise the security of the TOE, administrators are able to see what the attackers have been doing and thereby maybe reducing the damage done. This, of course, presumes that the attacker have not managed to change the security audit as well.

O.DATA_FLOW *For attached input/output devices a data flow control policy based on a threat analysis shall be made for each identified data flow. This is done to accommodate the different demands to secure communication of the devices.*

This security objective is identified to counter the threat T.DATA_FLOW. It assures that a threat analysis of each identified data flow is conducted in order to be able to make a data flow control policy. This ensures that the suitable level of protection of the data flows is implemented. The threat analysis and data flow control policy is discussed in section 2.2.3.

O.SESSION *A session shall only be active when an authorized user is interacting with the TOE interface. Therefore, the TOE shall provide functionality for the user to lock the current interactive session. It should also be possible for the TOE to automatically lock the session if the user is considered inactive. The user must re-authenticate to unlock the session. Furthermore, the user should re-authenticate before each sale and/or payment transaction.*

This objective is mainly identified to counter T.UNATTENDED_SESSION. By assuring that a session is only active when an authorized user is at the TOE interface the threat of an attacker taking advantage of an unattended session is eliminated. When the user leaves the TOE an automatic lockout procedure shall be initiated. This may be in form of a timeout interval or it can be by removing a smart card used for identification and authenticating. A session is in this context defined as the execution of a complete sales transaction including registration of goods and payments.

By demanding that the user shall re-authenticate before each sales transaction, P.ACCOUNTABILITY is also addressed because one operator can not overtake another operators session between sales transactions. This will be the case even if the TOE interface was not considered as unattended by the operator during the intervening period.

O.BACK-UP *The TOE shall provide functionality for administrators to back up the data in the system in order to make it possible to restore, as a minimum, the audit trail in case of hacking, hardware failure, fire, theft, force majeure, etc.*

This security objective is identified to counter T.PHYSICAL. It ensures that the TOE supports a backup mechanism so, at least, the audit trail can be restored in case of physical loss.

Also, if the audit trail is maliciously modified or deleted, e.g. in connection with unauthorized modification by an attacker, the security objective may be restored from the backup and by that countering T.MODIFICATION.

2.4.1 Security Objectives for the Environment

The above described security objectives are those to be complied with by the TOE itself. The security objectives described in this section are those to be complied with by the TOE environment. Normally, these objectives will be divided into security objectives for the IT and non-IT environment respectively. As the generalized model of the POS system is supposed to be self-contained as an IT system it will not have any security objectives for the IT environment. For the general non-IT environment the following security objectives are identified:

OE.TRAIN *The overall responsible for the TOE shall arrange training for all authorized users of the TOE including the administrators.*

This objective is mainly identified to address A.NO_EVIL and P.TRAIN. It assures that users, including administrators, are trained in secure use of the TOE and thereby staying competent. In addition, the objective counters T.INCOMPETENCE and thereby also T.UNATTENDED_SESSION as secure use of the TOE includes that a session shall not be be left unattended.

OE.ADMIN_ VETTING *The overall responsible for the TOE shall perform vetting of administrators to ensure that they are competent and non-hostile.*

This objective has been identified solely to address A.NO_EVIL. It assures that administrators are selected after thorough screening of candidates, thereby significantly reducing the risk of an administrator being incompetent and hostile.

OE.PHYSICAL *The TOE shall be physically protected in such a way that attackers cannot remove the TOE or parts of the TOE which are critical to the security of the TOE, or in other ways physically compromise the TOE and the data it contains, i.e. the audit trail, security attributes, etc.*

This objective has been identified solely to counter T.PHYSICAL. It assures that the TOE is physically protected, e.g. by keeping the audit trail in a locked and perhaps fire-proof location.

Table 2.1 illustrates the relations between assumptions, threats, and OSPs on one side, and the stated security objectives for the TOE and TOE environment on the other side. It demonstrates that the relations are internally consistent as required.

	O.IA	O.MANAGE	O.AUDIT	O.DATA_FLOW	O.SESSION	O.BACK-UP	OE.TRAIN	OE.ADMIN_VETTING	OE.PHYSICAL
A.NO_EVIL							x	x	
T.ACCESS	x	x	x						
T.MODIFICATION	x	x	x			x			
T.PHYSICAL						x			x
T.UNATTENDED_SESSION					x		x		
T.INCOMPETENCE							x		
T.DATA_FLOW		x		x					
P.AUTHORIZED_USERS	x	x							
P.ACCOUNTABILITY	x	x	x		x				
P.TRAIN							x		

Table 2.1: Assumptions, threats, and OSPs in relation to the security objectives.

2.5 Security Functional Requirements

The security functional requirements (SFRs) are a refinement of the security objectives into a set of requirements which ensures that the TOE can meet its security objectives. The following section describes the SFRs in order to clarify how these are chosen. All SFRs used in the PP are found in CC part 2 [CC204].

2.5.1 Organization of CC Part 2

CC part 2 is a catalog containing all the SFRs predefined in the CC. It might seem like a jungle to decide which requirements that should be specified in a PP, and it is a very time consuming process to explore and get familiar with the SFRs. It is particularly when studying the CC part 2, that other already published and evaluated PPs, will be of great help.

The CC offers 11 different classes, each containing a number of families. The classes each cover a general security subject such as *FAU — Security Audit, FIA — Identification and Authentication, FTA — TOE Access*, etc. A family covers a subject in the domain of the class such as *FAU_GEN — Security Audit Data Generation* in class FAU.

The families contain the functional components from which the security functional requirements of a PP (and ST) are build, e.g. *FAU_GEN.1 — Audit Data Generation* and *FAU_GEN.2 — User Identity Association* in the FAU_GEN family. Components can be leveled within the family such that one component is hierarchical to another. This means that the hierarchical component is an augmented version of the component it is hierarchical to. Components may also have dependencies on other components, even across the borders of classes and families. For instance *FAU_GEN.2 — User Identity Association* has dependency on *FAU_GEN.1 — Audit Data Generation* and *FIA_UID.1 Timing of Identification.* This dependency makes sense, as there must be audit records with which to associate users and these users must be identified. When a SFR component has a dependency on another component this shall be implemented as well.

The SFR components consist of one or more functional elements specifying the functional requirements. When taken directly from the catalog, most functional elements dictates assignments and/or selection operations to be carried out in order to specify the exact requirement it is to satisfy. Some elements also need to be refined to clarify phrasing or to change minor details element. If more significant changes are to be done a complete rephrasing of a SFR component is possible but then the component shall be renamed and handled as a explicit (custom) security requirement. Explicit SFR components may also be used to state requirements that are not covered by the predefined CC SFRs.

SFR components may also be iterated in order to use the same component with different assignments and selections. This could be when using the same component to state functional requirements to several objects or components of the TOE.

2.5.2 TOE Security Functional Requirements

This section describes the SFRs which are found suitable to satisfy the TOE security objectives. The SFRs used to satisfy the security objectives are listed in table 2.2. Some components are mentioned more than once because they address more than one objective.

Security Objective	Security Functional Requirement
O.IA	FIA_UAU.2 FIA_UAU.6 FIA_UID.2 FMT_SMF.1 FMT_SMR.1
O.MANAGE	FMT_MOF.1 FMT_MSA.1 FMT_MSA.3 FMT_MTD.1 FMT_SMF.1 FMT_SMR.1
O.AUDIT	FAU_GEN.1 FAU_GEN.2 FAU_SAR.1 FAU_SAR.2 FAU_STG.1 FMT_UID.2 FMT_SMF.1 FMT_SMR.1 FPT_STM.1
O.DATA_FLOW	FDP_IFC.1 FDP_IFF.1
O.SESSION	FIA_UAU.6 FTA_SSL.1 FTA_SSL.2
O.BACK-UP	FMT_SMF.1

Table 2.2: SFRs refining security objectives.

The following sections describes the TOE security functional requirements. The review is divided into functional areas corresponding approximatively to the TOE security objectives.

2.5.2.1 Identification and Authentication

Identification and authentication of users interacting with the TOE are essential to uphold security, as this and recording of security relevant events make it possible to hold users accountable for their actions within the TOE. The CC offers the class *FIA — Identification and Authentication* to serve this purpose. The families *FIA_UAU — User Authentication* and *FIA_UID — User Identification* provide components to identification and authentication.

The component *FIA_UAU.2 — User Authentication Before Any Action* is used to authenticate users. It states that before any actions is allowed, successful authentication of the user is required. FIA_UAU.2 is hierarchical to *FIA_UAU.1 — Timing of Authentication* which allows a specified list of actions even though the user is not authenticated. However, as users shall be held accountable for all their actions FIA_UAU.2 is chosen. The same arguments are used to clarify why *FIA_UID.2 — User Identification Before Any Action* is chosen to provide identification.

2.5.2.2 Data Flows

The CC has two different approaches to data protection. The class *FDP — User Data Protection* specifies components to protect user data and the class *FPT — Protection of the TSF* specifies components to protect TSF data. User data is data created by and for the user which does not affect the operation of the TOE. TSF data is data created by and for the TOE which might affect the operation of the TOE[4].

To protect the information in the data flows the first thing to determine is what type of data they contain. As the data contained in the financial part of the audit trail and the data contained in the data flows are created by a user, e.g. a an operator registering sales, it is concluded that the data flows need to be protected by the components in the FDP class.

This class offers two different approaches to protect the user data depending on the way the access to the data is controlled:

Discretionary Access Control (DAC) A means of restricting access to objects based on the identity and need-to-know of users and/or groups to which the object belongs. Discretionary access control systems permit users to entirely determine the access granted to their resources which means that they can through accident or malice give access to users which should have been unauthorized for that.

Mandatory Access Control (MAC) When using MAC, the user cannot fully control the access to resources that they create. The system security policy (as set by the administrator) entirely determines the access that is to be granted

[4] Definitions found in the glossary in [CC104]

and a user is not permitted to grant less restrictive access to their resources than the administrator specifies.

As the access control policies are to be defined by an administrator of the TOE the MAC approach is used. This means that *FDP_IFC.1 — Subset Information Flow Control* and *FDP_IFF.1 — Simple Security Attributes* shall be used to define the control policies to the user data. If the DAC approach was to be used the components *FDP_ACC.1 — Subset Access Control* and *FDP_ACF.1 — Security Attribute Based Access Control* should be implemented.

The component FDP_IFC.1, as it is described below, is refined to satisfy security objectives of the PP:

FDP_IFC.1.1 *The TSF shall enforce the [assignment: input/output device data flow control SFP] on [assignment: input/output devices which acts as TOE information interfaces and causes information to flow into and out of the audit trail.]*

This functional element requires that a uniquely named information flow control SFP (Security Function Policy), to be enforced by the TSF, is made for each identified input/output device data flow.

FDP_IFC.1 has dependency on FDP_IFF.1 as they will always follow each other. FDP_IFF.1 defines the actual input/output device data flow control policy and has six functional elements. Only the two first elements are described here. The other elements can be used to define additional rules and capabilities not included in the SFP.

FDP_IFF.1.1 *The TSF shall enforce the [assignment: input/output device data flow control SFP] based on the following types of subject and information security attributes: [Assignment: list of security attributes to be used to conduct a threat analysis of the data flow.]*

This functional element states that the TSF shall enforce a data flow control SFP on each input/output device. The SFP shall be based on a conducted threat analysis of the data flow based upon a list of security attributes. This relates directly to what was described in section 2.2.3.

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:*

a) A threat analysis of the input/output device data flow is carried out.

b) and the following countermeasures to achieve desired [selection: low, medium, high] level of protection for the data flow are implemented: [assignment: list of identified necessary countermeasures]

> This functional element states that a data flow is allowed if a threat analysis of the data flow has been carried out in order to determine the level of protection needed on the specific data flow and how to protect it. The threat analysis conducted is the one referred to in FDP_IFF.1.1.

FDP_IFF.1 has dependency on *FMT_MSA.3 — Static Attribute Initialization* described in section 2.5.2.5.

2.5.2.3 Audit

The CC offers the class *FAU — Security Audit* which specifies components to audit security relevant events in the TOE. When security auditing is implemented, the CC specifies which security relevant events to be audited for every SFR component throughout the CC part 2. The events are grouped as minimum, basic, or detailed level of audit depending on the importance of auditing the event. Other security relevant events than those predefined in CC part 2 may also be specified by the developer.

It is important to notice that this only addresses the *security* audit trail. The general term "audit trail" from the model defined in section 2.2.1 covers both the security audit trail and the financial audit trail. What to audit in the financial audit trail is a general functional requirement and not a security functional requirement. The financial part of the audit trail is handled as user data (see section 2.5.2.2). On the other hand, the security audit trail is handled as TSF data because it is generated by the TOE and influence on the security of the TOE.

The FAU_GEN family states requirements for recording appearance of security relevant events that take place under TSF control. *FAU_GEN.1 — Audit Data Generation* specifies requirements to recognize which auditable events that should generate audit records (FAU_GEN.1.1) and what information these records shall contain (FAU_GEN.1.2).

As users are held accountable for their actions within the TOE the component *FAU_GEN.2 — User Identity Association* is implemented. This component is a natural extension of FAU_GEN.1 as it gives the possibility to associate users with generated audit records. FAU_GEN.1 has dependency on *FPT_STM.1 — Reliable Time Stamps* which requires that the TSF provides reliable time stamps for TSF functions. FAU_GEN.2 has dependency on FAU_GEN.1 and *FIA_UID.1 — Timing of Identification* described in 2.5.2.1.

The above mentioned components only deal with audit generation. This means that components regarding audit review must be implemented as well. The family *FAU_SAR — Security Audit Review* specifies components for this purpose. The component *FAU_SAR.1 — Audit Review* gives authorized administrators the capabilities to read any audit information from the audit records (FAU_SAR.1.1). The audit records shall be presented in a way that makes administrators able to interpret these (FAU_SAR.1.2). FAU_SAR.1 has dependency on FAU_GEN.1. This makes sense as there must be audit records to review. *FAU_SAR.2 — Restricted Audit Review* requires that all other users than administrators shall be prohibited read access to the security audit records.

In order to uphold security, audit records may not be modified in any way by unauthorized users as this would allow attackers to cover their tracks. To prevent this, the component *FAU_STG.1 Protected Audit Trail Storage* is implemented. It assures that stored audit records are protected from unauthorized deletion (FAU_STG.1.1) and the TSF is able to prevent unauthorized modification of the audit records in the audit trail (FAU_STG.1.2).

2.5.2.4 Session

The security objective O.SESSION requires functionality to make it possible to lock a session, both manually by the user or automatically by the TOE. The class *FTA — TOE Access* provides a family *FTA_SSL — Session Locking* that provides components to address this.

As the TOE shall support both user and TOE initiated session locking the components *FTA_SSL.1 — TSF-Initiated Session Locking* and *FTA_SSL.2 — User Initiated Locking* are implemented.

FTA_SSL.1 assures that the TSF locks a session after the user has been inactive for a specific time frame. The locking is done by clearing display devices and disabling any other user functionality besides unlocking the session (FTA_SSL.1.1). The TSF shall require re-authentication in order to unlock the session (FTA_SSL.1.2).

FTA_SSL.2 assures that a user is able to lock the user's own session by clearing the display devices and disabling all other functionality besides unlocking of the session (FTA_SSL.2.1). In order to unlock the session the TSF shall require re-authentication (FTA_SSL.2.2). Both FTA_SSL.1 and FTA_SSL.2 have dependency on *FIA_UAU.1 — Timing of Authentication* described in section 2.5.2.1.

As it is required that a user is able to re-authenticate, the component *FIA_UAU.6 — Re-authenticating* is implemented. This specifies that the TSF shall re-authenticate users if a session has been locked, terminated, or a new sale or transaction are initiated.

2.5.2.5 Management

The selection of management components is more straightforward when the other functional requirements are stated because the management components required are often directly dependent on these. All management SFRs are collected in the class *FMT — Security Management.*

FDP_IFF.1 has dependency on *FMT_MSA.3 — Static Attribute Initialization.* It assures that the input/output device data flow control SFP is enforced by the TSF to provide restrictive default values for security attributes used by the SFP (FMT_MSA.3.1). In addition it states that the administrators are allowed by the TSF to override default values with specified alternative values (FMT_MSA.3.2). FMT_MSA.3 has dependency on *FMT_MSA.1 — Management of Security Attributes* and *FMT_SMR.1 — Security Roles.*

FMT_MSA.1 states that the TSF must enforce the input/output device data flow control SFP to restrict modification of security attributes in the relevant SFP to administrators.

FMT_SMR.1 states the roles to be maintained by the TSF and that users can be associated with these roles. The roles of the POS system is defined in 2.2.2 and can be assigned to the component directly. The roles are:

a) Customer

b) Operator

c) Financial Manager

d) Administrator

FMT_MOF.1 — Management of Security Functions Behavior allows only administrators to manage the behavior of the security functions of the TSF. The manageable functions are listed below.

a) The functions implementing the security auditing, including which security events to audit.

b) The functions implementing the input or output device data flow control policies for the attached input and output devices.

c) The functions implementing the method of identification and authorization of users.

d) The functions implementing timers and the clock synchronization.

e) The functions implementing the system backup routines.

f) The functions implementing the session locking methods.

In almost the same way the *FMT_MTD.1 — Management of TSF Data* component allows only the administrator to manage the TSF data created and protected by the TOE. The following data is manageable:

a) The security audit trail.

b) The TOE system clock.

Most of the FMT class SFRs has dependencies on *FMT_SMF.1 — Specification of Management Functions*. This component states which security management functions the TSF shall be able to perform. The other FMT components merely restricts the ability to manage the TOE. The identified management functions are listed below:

a) Functions to assign and maintain lists of users and roles.

b) Functions to create and recover backups of, as a minimum, the audit trail.

c) Functions to set up and manage information flow controls for input and output devices.

d) Functions to manage the TOE system clock and timers.

e) Functions to manage and review the security audit trail.

f) Functions to manage session locking attributes.

2.5.2.6 Backup

The only SFR component addressing the backup security objective is FMT_SMF.1 as described in section 2.5.2.5. It states that management functions to create and recover backups of, as a minimum, the audit trail is to be implemented. It does not, however, specify any other requirements for this function. No components from the CC part 2 has been found to specify backups and backup routines in more detail. Attempts have been carried out with combinations of e.g. *FDP_ROL Roll Back*, *FDP_ITC/FDP_ETC Import From/Export To Outside TSF control*, and the related families from the FPT class to include TSF data as well. However, no satisfying result was obtained.

A solution would be to define explicit SFRs to address O.BACKUP. This has not been carried out as it was considered to be too time consuming to fit within the time limits of PP development. Instead it is left to be specified in future iterations of the PP development.

2.6 Security Assurance Requirements

Assurance is the foundation for confidence in the fact that the IT product is meeting its security objectives. Assurance in CC context is provided through an evaluation of the IT product in order to determine its security attributes. The level of assurance is based on the evaluation effort.

2.6.1 Organization of CC Part 3

Part 3 of the CC [CC304] is a catalogue of assurance requirements which form the basis for the evaluation. It is organized in the same way as part 2 of the CC [CC204] in terms of classes, families, and components, see section 2.5.1 for details. However, it differs on component level where each assurance element is identified as belonging to one of three sets of assurance elements listed below.

1. Developer Action Elements which are actions that shall be performed by the developer, e.g. the developer shall provide user guidance. These are marked with the letter "D" appended to the element number, e.g. ADV_FSP.1.1D.

2. Content and Presentation of Evidence Elements which are the evidence required, what the evidence shall demonstrate, and what information the evidence shall convey. These are marked with the letter "C" appended to the element number, e.g. ADV_FSP.1.1C.

3. Evaluator Action Elements which are actions the evaluator shall perform. These are marked with the letter "E" appended to the element number, e.g. ADV_FSP.1.1E.

2.6.2 Assurance Security Requirements

Each assurance element represents an assurance requirement to be met. Security assurance requirements are derived from the chosen Evaluation Assurance Level (EAL) which is decided on the basis of the following assessment.

As the protection profile has been developed for an environment with moderate level of risk, it is therefore concluded that the level of assurance provided by EAL3 with no augmentation is appropriate. EAL3 gives a moderate level of independently assured security with a thorough investigation of the TOE and its development without incurring substantial re-engineering costs. Compared to EAL2, EAL3 gives more confidence in the fact that the TOE will not be tampered with during development. EAL3 includes the assurance components listed in table 2.3.

Class	Component	
ACM	ACM_CAP.3	Authorization controls
	ACM_SCP.1	TOE CM coverage
ADO	ADO_DEL.1	Delivery procedures
	ADO_IGS.1	Installation, generation and start-up procedures
ADV	ADV_FSP.1	Informal functional specification
	ADV_HLD.2	Security enforcing high-level design
	ADV_RCR.1	Informal correspondence demonstration
AGD	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
ALC	ALC_DVS.1	Identification of security measures
ATE	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: high-level design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
AVA	AVA_MSU.1	Examination of guidance
	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.1	Developer vulnerability analysis

Table 2.3: Security assurance components in EAL3.

2.7 PP Conclusion and Discussion

A Protection Profile for POS systems has been developed. The contents of the PP complies with the CC specification of PPs outlined in CC part 1 [CC104]. EAL3 has been chosen as the appropriate assurance level.

A general model of a POS system has been introduced in order to define the TOE and to address any type of POS system.

The PP has been defined in a way that enables TOEs, which are only a part or component of an entire POS system, to claim conformance to the PP. An interface model for POS devices has been introduced to support this.

The defined general POS system model is particularly concerned about the usage of POS devices and the input/output data flows they cause. Emphasis on these have been made because the main purpose of the PP is to define a POS system environment for later usage when developing a ST for a POS device. The model is, however, found applicable although other developers with different approaches might emphasize different features. This is the case for almost any other PP as well, because the security requirements will be subjective although the CC in general should be objective and formal. Hence, PPs for similar product types, e.g. fire walls, may be defined differently depending on the approach of the developer.

The CC does not support partial PP conformance. This cause a great obstacle when writing PPs for product types which are composed by several parts and components developed be different providers, e.g. POS systems. The PP counters this problem by stating that conformance claims are possible if objectives and related TSF requirements, not directly addressed by the TOE, are moved to the TOE IT environment, i.e. the general IT POS environment. Nothing in the CC indicates that this construct should conflict with the CC and it is therefore assumed to be an accepted solution.

Topics to be considered more thoroughly in future versions of the PP may include the security requirements satisfying the O.BACK-UP security objective. The present PP only requires that back-up functionality is implemented but is does not state any security functional requirements. This is because no relevant TSFs were found specifically addressing backup. However, explicitly stated SFR components may be included in order to do that in future versions of the PP.

CC part 3 may also be subject to a more thorough examination in order to find possible relevant assurance requirements which might augment the evaluation assurance level stated. Enough time have probably not been put into this as the security functional requirement have had higher priority.

When developing a PP it is important to have in mind that the security requirements stated are kept within a realistic scope. If a PP is developed with too extensive requirements, no ST will be able to claim conformance and the PP will be useless, hence no additional security is obtained. When developing the PP this have been an important issue and it is found that a realistic set of requirement has been stated.

CHAPTER 3

Security Target

This chapter describes the development of the Security Target (ST). The ST is attached as appendix B and it is recommended to read this simultaneously with the chapter as sections of the ST are walked through chronologically.

3.1 ST Development

This section describes the approach on how to develop a ST.

3.1.1 Contents of a Security Target

The contents of a ST is very similar to that of a Protection Profile which is described in 2.1.1. However, there are a few differences. A ST contains two extra main sections. These are added between the *IT Security Requirements* section and the *Rationale* section and are described below.

1. TOE Summary Specification — This section describes the security functions and assurance measures of the TOE that meet the TOE security and assurance requirements.

2. PP Claims — This section shall be only included if the ST claims conformance to a PP. It shall provide the documentation necessary to explain how the conformance claims are met.

Additionally, the Rationale section is extended with rationales for the TOE Summary Specification and the PP Claims in order to prove that the ST is internally consistent.

The organization of a ST is, like the PP, very intuitive. The TOE Summary Specification is an informal specification of TOE security functions and assurance measures. This may serve as a first step in the design phase of the IT system and is therefore a natural next step in the development process. The PP claims, if any, are a in order to justify the conformance claims made.

3.1.2 Literature

As mentioned in section 2.1.2 not much literature, other than parts 1-3 of the CC, can be found on the Common Criteria including how to write a ST. But these are not very helpful to the inexperienced ST developer. However, the Common Criteria web site[1] offers a comprehensive list of evaluated STs and inspiration among these can be sought.

The *Windows 2000 Security Target* [Mic02] covers the Windows 2000 operating system. It claims conformance to the *Controlled Access Protection Profile* (CAPP) [Inf99] which was derived from the requirements of the C2 class of the Trusted Computer System Evaluation Criteria (TCSEC). These have been a great help when stating the PP conformance claims and the PP tailoring.

The *Security Target for Citrix MetaFrame XP Presentation Server for Windows with Feature Release 3* [Cit04] which, in general terms, describes how to provide a secure

[1] www.commoncriteria.org

data flow between two applications over an insecure medium. This ST share similar functionalities with the ST to be developed and has been an inspiration in defining the requirements to protect the TOE.

3.2 The ST TOE

The goal is to develop a Common Criteria Security Target for a secure POS application interface for POS payment terminals when the communication between the POS application and a payment terminal is to be transmitted over an insecure media, e.g. the Internet. This is the case when integrating payment terminals in hosted POS systems where the Internet is used for communication between the POS application server and POS client, see section 1.5.6. The purpose of the TOE is to protect the communication by providing a trusted channel between the POS application interface and the payment terminal interface.

The ST shall be able to claim conformance to the PP for POS systems version 1.0 defined in appendix A. As the TOE obviously is not an entire POS system but merely a part or component of one, special conformance claims as described in section 2.2.4 must be utilized.

3.2.1 TOE Model

When referring to the generalized POS system model defined in section 2.2, the payment terminal is a POS device generating data flows in and out of the audit trail. If the POS system was a non-hosted type, the payment terminal could have been interfaced directly with the POS application and the interface model would look like the general model as illustrated in figure 2.2 section 2.2.3. Because the POS application interface and the device driver interface are physically and logically separated, they cannot be interfaced directly. The TOE will act as a bridge between the two interfaces and at the same time protect the communication. Figure 3.1 illustrates this interconnection.

The gray areas indicate the TOE components and the interfaces marks the boundaries between the TOE and the POS IT environment. This distinction of TOE and

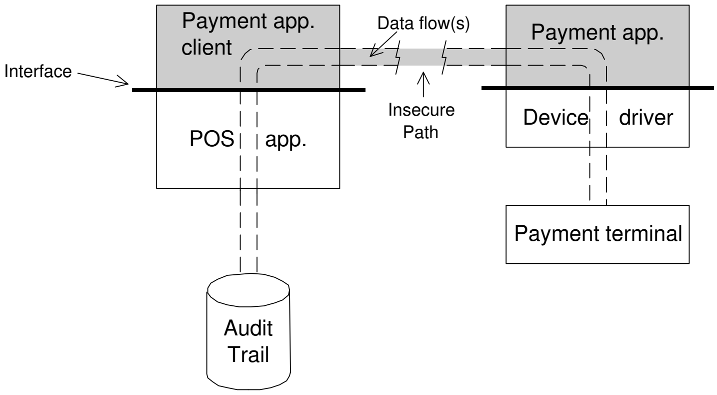

Figure 3.1: Interconnection of POS application interface and device driver interface.

POS IT environment is essential when special conformance claims, as described in section 2.2.4, is needed.

The TOE consists of two main components: The Payment Application (PA) and the Payment Application Client (PAC).

The PA The payment application implements the payment device driver interface of the TOE, hence it is located at the actual point of sale and most likely running on the POS terminal. The PA receives commands from the merchant via the PAC and transmits the responses back to the PAC when the commands are executed. In addition, the PA shall monitor and handle any other event raising from the device driver, i.e. the payment terminal, and transmit these to the PAC when this is connected during a transaction.

The PAC The payment application client implements the POS application interface of the TOE, hence it is located at the POS application and running on the POS application server. The POS application initiates the PAC whenever the merchant wants to perform a transaction on the payment terminal. The PAC then initiates a trusted communication channel between itself and the PA to transmit the commands, responses and events the transaction will give rise to. When the transaction is finished, the PAC shall terminate the connection.

3.2.2 The Payment Terminal

The payment terminal is assumed to be compliant with the Open Terminal Requirement Specification (OTRS) [PBS04]. OTRS is the Danish functional and security requirements specification for the new chip card enabled payment terminals which have been introduced to the Danish market over the last few years. These payment terminals are also refereed to as “Flex Terminals”. OTRS is specified and maintained by PBS A/S[2], who is presently the only payment terminal operator on the Danish market. Although the OTRS is a Danish specification, it is based on the Terminal Architecture for PSAM Applications (TAPA) [Eur01], which is an international standard jointly developed by Europay, VISA and PBS.

The Flex Terminal have several important features which differentiates them from the older payment terminal models. The most important is the TAPA architecture, which introduces the PSAM (Purchase Secure Application Module). The PSAM is a module controlling the general functionality and security policies of the terminal and it is implemented by the terminal operator (PBS).

The Flex Terminals support both Magnetic Stripe Cards (MSC) and Integrated Circuit Cards (ICC). Authorization may be performed both online and off line and Card Holder Verification (CVM) may be performed by signature or PIN entry. ICCs may

[2] `www.pbs.dk`

also contain several applications, each implementing one debit/credit card scheme, e.g. VISA debit card and Mastercard credit card. The security policies in the PSAM defines which combinations of card type (MSC/ICC), method of authentication (online/offline), CVM, and card scheme is accepted at the individual payment terminal.

3.2.3 Roles

The PP defines four roles of users of a POS system: customer, operator, financial manager, and administrator. Before defining the input/out device data flows an additional terminal operator role must be introduced:

Terminal Operator This role will be the source and destination role for the data flows to and from the payment terminal. As described before, the terminal operator controls the PSAM, which then again controls the general functionality and security policies of the payment terminal.

3.2.4 Data flows

As stated in the PP section A.2.3 the input/output device data flows of the payment terminal must be identified in order to perform a threat analysis of these. The threat analysis is the foundation of the input/output device data flow control SFP.

The data flows identified are derived from the functional requirements of the OTRS. Any data flow will fall into one of the following types

- Commands
- Responses
- State information
- Terminal requests

Commands may be transactional or administrative depending their purpose. Transactional commands are supposed to be executed by the operator and the administrative commands are executed by the financial administrator. The identified data flows are listed below:

Transactional Commands These are the operator initiated commands which initiates a payment transaction, make it change state, or terminate. When initiating a payment transaction the desired amount and currency, type of authorization, and CVM is transmitted to the terminal. If no type of authorization or CVM is stated, the PSAM (and payment card in case of a ICC) selects a default best choice as stated in the security policy of the PSAM. If the transaction has not yet been authorized it may be terminated by a “cancel” or “clear” command which are also categorized as transactional commands.

Transactional Command Responses The responses of a command is first of all whether or not the command was successfully send to the terminal and, when the command has been executed, the result of the command. The result of a transactional command includes whether or not the transaction was authorized and a receipt documenting the transaction with respect to time, amount, card PAN, CVM, PSAM number, etc. The receipt shall be printed out and handed over to the customer by the operator. Depending on the type of transaction an extra receipt may be required, e.g. when performing refunds and signature authenticated transactions.

Administrative Commands These are commands initiated by the financial manager to change the state of the payment terminal, flush data stores, or request batch reports for totaling and financial accounting. Flushing of data stores is to be done at a daily basis to ensure that stored transactions are delivered to the terminal acquirer in time. Stored transactions occur when off line transaction are performed.

Administrative Command Responses Responses to administrative commands include whether or not the command was successfully initiated and the result of the command. When requesting batch reports these will be part of the resulting data.

State Information Messages These messages inform the operator about the current state of the payment terminal. State information messages may be trivial information like “Waiting for card”, “Closed”, and “Approved”. The latter only as supplementary information to the actual transaction command response. However, sometimes the operator may need to act upon a message, i.e. “Try again”, “Recovery needed”, and “Suspected fraud”. The complete list of state information messages is listed in the OTRS.

Terminal Requests As the Flex Terminals support many different transaction types in relation to CVM, authorization methods, etc., the PSAM may need to explicitly request the operator to perform some kind of action and respond to this. This is e.g. the case when:

- Performing signature transaction when the operator is asked to validate the card holder signature.
- Performing an ICC transaction and the card reader is unable to read the chip on the card. The operator is then asked to verify that the card is correctly inserted in order to enable a MSC transaction as fall-back (MSC fall-back).
- Performing an ICC transaction and the card contains several applications, i.e. card schemes. The operator may then be asked to assist the card holder in selecting the desired application.

- Performing an off line transaction the operator may be asked to check the card PAN against a stop list, possibly by calling the acquirer by phone and receive a “voice approval code”.

Terminal Request Responses These are the responses from the operator to a terminal request from the PSAM. The data may include the voice approval code or the selected card application depending on the type of request.

3.2.4.1 Protection of Data Flows

In order to protect the data flows a threat analysis must be carried out. The threat analysis shall lead to the definition of the level of protection required, and the countermeasures to uphold this shall be defined. This, altogether, defines the input/output device data flow control SFP of the payment terminal. This SFP is identified as the *Payment Terminal Data Flow Control SFP*.

The PP states the following attributes to be considered when performing the threat analysis:

- Type of input/output device used in the data flow.
- Role of user creating and receiving the data.
- Type and sensitivity of the data.
- Media in which the data flows.
- Possible threat agents.

These attributes are considered adequate for the threat analysis, hence no additional attributes are defined. The threat analysis is described in section B.2.4 of the ST. The conclusion of the threat analysis is that the data flows shall be protected by countermeasures providing a high level of protection. This is due to the combination of high sensitivity of the data and the insecure media in which the data is transmitted. Even though seven data flows are identified, only one data flow control policy is defined covering them all. This is because the data flows are almost similar in relation to the sensitivity of data, and this will ease the final implementation.

The countermeasures to be implemented to provide the high level of protection shall use strong cryptographic functions to implement confidentiality and integrity of the data flows as well as mutual authentication of the PA and the PAC.

3.3 TOE Security Environment

This section describes the TOE security environment. Opposed to section 2.3 this section only describes modifications relative to the PP, i.e. if assumptions, threats, organizational security policies (OSPs) have been added or enhanced. Table 3.1 shows modifications to assumptions, threats, and OSPs relative to the PP.

Name	Modification
A.NO_EVIL	None
A.THIRD_PARTY	Added
T.ACCESS	None
T.MODIFICATION	None
T.PHYSICAL	None
T.UNATTENDED_SESSION	None
T.INCOMPETENCE	None
T.DATA_FLOW	Enhanced
T.AUTHENTIC	Added
T.CRYPTO_KEY	Added
P.AUTHORIZED_USERS	None
P.ACCOUNTABILITY	None
P.TRAIN	None
P.FIPS	Added

Table 3.1: Modifications to assumptions, threats, and OSPs relative to the PP.

3.3.1 Assumptions

A new assumption, A.THIRD_PARTY which is stated below, has been added relative to the PP. This assumption has been introduced in order to make sure that all third party products are assumed to be trusted. This includes operating systems, cryptographic service providers (CSPs), etc. If this assumption is not made the environment might be insecure and thereby compromising the security of the TOE.

A.THIRD_PARTY *It is assumed that all third-party products used to implement the TOE environment (the general POS system, cryptographic service providers, etc) are trusted as well as correctly installed and configured.*

3.3.2 Threats to Security

As a ST is developed for a specific product, new threats may arise. Therefore, it may be necessary to enhance some threats or add new threats compared to the PP. The following sections explains these enhancements and additions.

3.3.2.1 Threats

T.DATA_FLOW *An attacker may compromise the confidentiality and integrity of an input or output data flow.*

T.DATA_FLOW has been enhanced in the sense that a threat against *confidentiality* has arisen as the threat now addresses the identified data flows in the TOE which contain sensitive data, see section 3.2.4.

T.AUTHENTIC *An attacker may try to impersonate the payment application or client, e.g. redirect the client connection to an unauthentic application.*

This threat is introduced as it is a security risk if a PAC is able to connect to an unauthenticated PA or if an unauthenticated PA is able to connect to the PA. This might lead to disclosure of cryptographic keys, unauthorized access to audit trail, etc.

T.CRYPTO_KEYS *An attacker may compromise the security of the TOE by disclosing cryptographic keys in the TOE.*

This threat is introduced as disclosure of cryptographic keys will compromise the security of the TOE. If an attacker obtains the cryptographic keys, confidentiality and integrity of the data flows will be compromised.

3.3.3 Organizational Security Policies

P.FIPS140 *Any cryptographic function used by the TOE shall be FIPS 140 level 1 compliant.*

This policy has been introduced to require that all cryptographic functions shall be FIPS 140 level 1 validated in order to ensure that the cryptographic algorithms use the function as intended. FIPS 140 level 1 is considered adequate as higher FIPS 140 levels make requirements for physical security.

3.4 Security Objectives

This section describes the security objectives stated in the ST. Compared to the PP some of the security objectives have been moved to the IT environment and new objectives are stated in order to counter new threats and address new assumptions and OSPs. Table 3.2 shows which objectives that remain in the TOE environment, which are moved to the IT environment, and new security objectives. The following sections describe why the new objectives have been added and why some objectives are moved.

Name	**Modification**
O.MANAGE	None
O.AUDIT	None
O.DATA_FLOW	None
O.AUTHENTIC	Added
O.TRUSTED_CHANNEL	Added
O.IA	Moved
O.SESSION	Moved
O.BACK-UP	Moved
O.FIPS140	Added
OE.TRAIN	None
OE.ADMIN_VETTING	None
OE.PHYSICAL	None

Table 3.2: Modifications relative to the PP.

3.4.1 Added Security Objectives

O.AUTHENTIC *The payment application and payment application client shall perform mutual authentication before allowing any communication.*

This objective has been added mainly to counter T.AUTHENTIC as it requires that the payment application (PA) and the payment application client (PAC) perform mutual authentication before any communication is allowed.

In addition it counters T.CRYPTO_KEYS as the risk of disclosure is minimized when the two parties are mutually authenticated before cryptographic keys are shared. T.DATA_FLOW is countered as well because a data flow cannot be initiated unless the PA and PAC are mutually authenticated. This minimizes the risk of loss of confidentiality and integrity of the data flow. T.ACCESS is countered indirectly, as connection to an unauthenticated application is impossible, which reduces the risk of unauthorized access to the TOE.

O.TRUSTED_ CHANNEL *The authentication process, session key distribution* *and communication of sensitive data shall be protected by a trusted channel between the payment application and payment application client.*

This objective is mainly added to counter T.DATA_FLOW. It requires establishment of a trusted channel between the PA and the PAC, thereby protecting the data flows.

In addition it counters T.AUTHENTIC and T.CRYPTO_KEYS as the initialization of the trusted channel includes mutual authentication of the two parties and secure exchange of cryptographic keys.

O.FIPS140 *The cryptographic service providers used to provide the cryptographic functions for the TOE shall be FIPS 140 validated to, at least, level 1.*

This objective has mainly been identified to address P.FIPS140 in order to assure that all cryptographic functions are FIPS 140 level 1 validated.

In addition it counters T.DATA_FLOW, T.AUTHENTIC, and T.CRYPTO_KEYS indirectly through the requirement that the trusted channel is established by validated cryptographic functions. It addresses A.THIRD_PARTY as it ensures that cryptographic functions in third party products shall be FIPS 140 level 1 validated as well.

3.4.2 Security Objectives Moved to the Environment

Some of the objectives stated in the PP cannot be complied by the TOE and are therefore moved to the environment in order to be able to claim conformance to the PP, see section A.6 for details.

The POS IT system provides access control to the functionality of the POS system. The TOE access control is enclosed in this and therefore the security objectives regarding TOE access, O.IA and O.SESSION, are moved to the environment.

As the audit trail is maintained by the POS IT system, it is the POS IT system which is able to gain access to the audit trail in order to backup or recover this. Therefore O.BACKUP is moved to the environment.

In special cases security objectives may be satisfied by both the TOE and the environment. For instance, audit records are generated by the TOE while the POS IT system maintains the audit trail and functionality to review this. Section 3.5 describes this.

The general environment has not changed and therefore security objectives regarding this are left unchanged.

Table 3.3 illustrates the relations between assumptions, threats, and OSPs on one side, and the stated security objectives for the TOE and TOE environment on the other side. It demonstrates that the relations are internally consistent as required.

	O.MANAGE	O.AUDIT	O.DATA_FLOW	O.AUTHENTIC	O.TRUSTED_CHANNEL	O.IA	O.SESSION	O.BACK-UP	O.FIPS140	OE.TRAIN	OE.ADMIN_VETTING	OE.PHYSICAL
A.NO_EVIL										x	x	
A.THIRD_PARTY									x	x	x	
T.ACCESS	x	x		x		x						
T.MODIFICATION	x	x				x		x				
T.PHYSICAL								x				x
T.UNATTENDED_SESSION							x			x		
T.INCOMPETENCE										x		
T.DATA_FLOW	x		x	x	x				x			
T.AUTHENTIC				x	x				x			
T.CRYPTO_KEYS				x	x				x			
P.AUTHORIZED_USERS	x					x						
P.ACCOUNTABILITY	x	x				x	x					
P.TRAIN										x		
P.FIPS140									x			

Table 3.3: Assumptions, threats, and OSPs in relations to the security objectives.

3.5 Security Functional Requirements

This section describes the Security Functional Requirements (SFRs) stated in the ST. The SFRs used to satisfy the security objectives are shown in table 3.4. Security objectives and SFRs written in **bold** font are new relative to the PP.

Security Objective	**Security Functional Requirement**
O.MANAGE	FMT_MOF.1 FMT_MSA.1 **FMT_MSA.2** FMT_MSA.3 FMT_MTD.1 FMT_SMF.1 FMT_SMR.1
O.AUDIT	FAU_GEN.1 FAU_GEN.2 FAU_SAR.1 FAU_SAR.2 FAU_STG.1 FMT_UID.2 FMT_SMF.1 FMT_SMR.1 FPT_STM.1
O.DATA_FLOW	**FCS_CKM.1** **FCS_CKM.2** **FCS_CKM.4** **FCS_COP.1** FDP_IFC.1 FDP_IFF.1 **FDP_ITT.1** **FDP_ITT.3** **FPT_ITT.1** **FTP_ITC.1**
O.AUTHENTIC	**FDP_ITT.1** **FDP_ITT.3** **FPT_ITT.1** **FTP_ITC.1**
O.TRUSTED_CHANNEL	**FCS_CKM.1** **FCS_CKM.2** **FCS_CKM.4** **FCS_COP.1** **FTP_ITC.1**

Continued on next page.

Security Objective	Security Functional Requirement
O.IA	FIA_UAU.2 FIA_UAU.6 FIA_UID.2 FMT_SMF.1 FMT_SMR.1
O.SESSION	FIA_UAU.6 FTA_SSL.1 FTA_SSL.2
O.BACK-UP	FMT_SMF.1
O.FIPS140	**FCS_COP.1** **FCS_CKM.1** **FCS_CKM.2** **FCS_CKM.4**

Table 3.4: How the SFRs are derived from security objectives.

As new security objectives are stated relative to the PP, new SFRs are derived from these. As some of the security objectives have been moved to the environment the SFRs satisfying these are moved to the environment as well. SFRs satisfying the TOE have been assigned specific values while SFRs satisfying the POS IT environment are left open.

As mentioned in section 3.4.2 security objectives may be satisfied by both the TOE and the environment. This is done by iterating the component used to satisfy the objective. Functionality provided by the TOE will be placed in the TOE SFR iteration and functionality provided by the POS IT system will be placed in the environment iteration. Table 3.5 shows these modifications.

3.5.1 Identification and Authentication

FIA_UAU.2 — User Authentication Before Any Action and *FIA_UID.2 — User Identification Before Any Action* are both moved to the IT environment as the security objective from which they are derived, O.IA, is moved to the IT environment.

3.5.2 Data Flows

Compared to the PP the components *FDP_IFC.1 — Subset Information Flow Control* and *FDP_IFF.1 — Simple Security Attributes* have now been assigned specific values to identify and define the Payment Application Data Flow Control SFP. Below the elements with assignments are stated and described.

FDP_IFC.1.1 *The TSF shall enforce the Payment Application Data Flow Control SFP on data flowing between the payment terminal and the POS application which causes information to flow into and out of the audit trail.*

Class	**Modification**
FAU_GEN.1	Assignment Iterated
FAU_GEN.2	Moved
FAU_SAR.1	Moved
FAU_SAR.2	Moved
FAU_STG.1	Moved
FCS_CKM.1	New
FCS_CKM.2	New
FCS_CKM.4	New
FCS_COP.1	New
FDP_IFC.1	Assignment
FDP_IFF.1	Assignment
FDP_ITT.1	New
FDP_ITT.3	New
FIA_UAU.2	Moved
FIA_UAU.6	Moved
FIA_UID.2	Moved
FMT_MOF.1	Assignment Iterated
FMT_MSA.1	Assignment
FMT_MSA.2	New
FMT_MSA.3	Assignment
FMT_MTD.1	Assignment Iterated
FMT_SMF.1	Assignment Iterated
FMT_SMR.1	Moved Assignment
FPT_ITT.1	New
FPT_STM.1	Moved
FTA_SSL.1	Moved
FTA_SSL.2	Moved
FTP_ITC.1	New

Table 3.5: Modifications to SFRs relative to the PP.

Now it is explicitly stated that the Payment Application Data Flow Control SFP shall be enforced on the data flows between the payment terminal and the POS application. All identified data flows included in the SFP are described in section 3.2.4.

FDP_IFF.1.1 *The TSF shall enforce the Payment Application Data Flow Control SFP based on the following types of subject and information security attributes:*

a) Type of input/output device used in the data flow.

b) Role of user creating and receiving the data.

c) Type and sensitivity of the data.

d) Media in which the data flows.

e) Possible threat agents.

Here it is stated that the Payment Application Data Flow Control SFP shall be enforced on the basis of a threat analysis based on the listed 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:*

a) A threat analysis of the input/output device data flow is carried out.

b) and the following countermeasures to achieve desired "high" level of protection for the data flow are implemented:

1) Secure authentication between the payment application and client ensuring correct authorization of the end points.

2) Encryption of the data flow using 3DES or AES ensuring the confidentiality and integrity of the data.

This functional element states that a data flow can only be allowed if a threat analysis of the data flow has been conducted and counter measures to ensure a high level of protection of the data flow is implemented. The threat analysis conducted in section B.2.4 concluded that a high level of protection is needed due to the sensitivity of the data.

Therefore, the TOE shall be able to provide mutual authentication between the PA and PAC ensuring that the PAC is communicating with the authentic PA. Furthermore, the data flow shall be encrypted using 3DES or AES with appropriate key lengths.

In order to secure the data flows a trusted channel between the PA and PAC shall be implemented. In order to achieve this the class *FTP — Trusted Path/Channel* is examined. This class provides the family *FTP_ITC — Inter-TSF Trusted Channel* which defines requirements for creation of a trusted channel for secure communication.

When the trusted channel is established communication between the PA and PAC will be considered as internal TOE transfer instead of inter-TSF transfer[3]. To further strengthen the requirements to the security of the data flows the components

[3] See figure 1.2 [CC204] p. 4.

FDP_ITT.1 — Basic Internal Transfer Protection and *FPT_ITT — Internal TOE Data Transfer* are introduced as both user and TSF data are transferred in the trusted channel. FDP_ITT.1 enforces the Payment Application Data Flow Control SFP to prevent modification and disclosure of user data transferred between the PA and PAC. FPT_ITT.1 protects TSF data from disclosure and modification when data is transmitted between the PA and the PAC.

As the trusted channel makes use of cryptographic functions, securing communication, cryptographic support must be implemented. The class *FCS — Cryptographic Support* provides this. It contains the two families *FCS_ CKM — Cryptographic Key Management* and *FCS_ COP — Cryptographic Operation.*

FCS_ COP.1 — Cryptographic Operation is used to define which cryptographic operations the TOE shall support in order to implement encryption of the data flows and mutual authentication of the PA and PAC. It states that the TLS protocol shall be used to implement these functions with one of the following TLS cipher suites as described in [DA99] and [Cho02]:

a) TLS_RSA_WITH_3DES_EDE_CBC_SHA,

b) TLS_RSA_WITH_AES_128_CBC_SHA, or

c) TLS_RSA_WITH_AES_256_CBC_SHA.

FCS_COP.1 has dependency on *FCS_ CKM.1 Cryptographic Key Generation* and *FCS_ CKM.4 Cryptographic Key Destruction.*

FCS_CKM.1 requires that the TOE shall generate cryptographic keys in accordance with a Secure Hash Standard based (SHS) random number generation as specified in FIPS 186 [U.S00] appendix 3 or an equivalent SHS based algorithm. Symmetric key sizes must be in accordance with the ones specified in FCS_COP.1 above. FCS_CKM.4 requires that the TOE shall destroy cryptographic keys using any FIPS 140 level 1 validated key destruction method.

As keys are exchanged between the PA and the PAC a cryptographic key distribution method shall be stated. *FCS_ CKM.2 — Cryptographic Key Distribution* requires that cryptographic keys shall be distributed using RSA based key exchange, given by the TLS cipher suites, complying with FIPS 140.

FCS_COP.1, FCS_CKM.1, FCS_CKM.2, and FCS_CKM.4 all have dependency on *FMT_ MSA.2 — Secure Security Attributes* which is described in section 3.5.5.

To ensure integrity of the data flows the component *FDP_ITT.3 — Integrity Monitoring* is implemented. It requires the TOE to enforce the Payment Application Data Flow Control SFP to monitor the transmitted data for cryptographic integrity errors. If errors are detected data shall be attempted to be resend a specified number of times before alerting the administrator.

3.5.3 Audit

The component *FAU_GEN.1 — Audit Data Generation* has been assigned in order to require that all auditable events for the detailed level of audit are recorded. The component is also iterated as security relevant events may occur in the environment.

FAU_GEN.2 — User Identity Association is moved to the environment as it is the POS IT system which handles user identification and authentication, hence it is the POS IT system which is able to associate a user with an audit record. *FPT_STM.1 — Reliable Time Stamps* is also moved to the environment as the TOE shall make use of the POS system clock when creating audit records. *FAU_SAR.1 — Audit Review*, *FAU_SAR.2 — Restricted Audit Review*, and *FAU_STG.1 — Protected Audit Trail Storage* are all moved as well because it is left to the IT environment to provide functionality to review audit records and to store the audit trail in a secure way.

3.5.4 Session

The components *FTA_SSL.1 — TSF-Initiated Locking*, *FTA_SSL.2 — User Initiated Locking*, and *FIA_UAU.6 — Re-authenticating* are all moved to the environment as the security objective from which they are derived, O.SESSION, has been moved to the environment.

3.5.5 Management

FMT_SMR.1 — Security Roles which specifies the roles maintained has been assigned a new role, *Terminal Operator* described in section 3.2.3. The roles are still maintained by the POS system and the component is therefore moved to the environment.

FMT_MSA.1 — Management of Security Attributes and *FMT_MSA.3 — Static Attribute Initialization* have both been assigned to enforce the *Payment Application Data Flow Control SFP*.

FMT_MOF.1 — Management of Security Functions Behavior, *FMT_MTD.1 — Management of TSF Data*, and *FMT_SMF.1 — Specification of Management Functions* have all been iterated in order to specify which functions should be maintained by the TOE and the environment respectively. The management functions of the TOE only counter manageable functionality and TSF data relating to audit generation and the cryptographic operations. The management functions of the POS IT environment counter any other management functionality described in the PP.

FMT_MSA.2 — Secure Security Attributes is introduced because the four cryptographic components all have dependency on it. FMT_MSA.2 requires that only secure values are accepted as secure attributes, e.g. that cryptographic keys have the right key length.

3.5.6 Backup

As the security objective O.BACKUP has been moved to the environment the requirement for backup functions stated in FMT_SMF.1 has been moved to the component iteration in the environment.

3.6 Evaluation Assurance Level Selection

This section describes how the security assurance requirements are chosen for the ST. *EAL3 — Methodically Tested and Checked* was found suitable to cover the assurance needs for the PP, see section 2.6 for details.

Even though sensitive data is now transported in the TOE the general threat scenario has not changed. Therefore EAL3 is still adequate to assure the security of the TOE. However, FMT_MSA.2 has dependency on ADV_SPM.1. Therefore, this component has been added to the security assurance requirements and the assurance level is upgraded to EAL3+. Included assurance requirements are listed in table 3.6.

Class	**Component**	
ACM	ACM_CAP.3	Authorization controls
	ACM_SCP.1	TOE CM coverage
ADO	ADO_DEL.1	Delivery procedures
	ADO_IGS.1	Installation, generation and start-up procedures
ADV	ADV_FSP.1	Informal functional specification
	ADV_HLD.2	Security enforcing high-level design
	ADV_RCR.1	Informal correspondence demonstration
	ADV_SPM.1	Informal TOE security policy model
AGD	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
ALC	ALC_DVS.1	Identification of security measures
ATE	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: high-level design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
AVA	AVA_MSU.1	Examination of guidance
	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.1	Developer vulnerability analysis

Table 3.6: Security assurance components.

3.7 TOE Summary Specification

This section describes the TOE summary specification which defines the TOE security functions and assurance measures. Depending on how the TOE summary specification is written the functional requirements may be quite similar, or even identical, to the information to be provided for the TOE as part of the *ADV_FSP — Informal Functional Specification*.

3.7.1 TOE Security Functions

The following TOE security functions have been identified: F.CRYPTOGRAPHIC, F.AUDIT, and F.MANAGEMENT. The statements of the security requirements can be found in section B.6.1.

The TOE security functions are derived from the TOE security objectives and reflects the security functionality of the TOE. F.AUDIT is derived from O.AUDIT, F.CRYPTOGRAPHIC is derived from O.DATA_FLOW, O.TRUSTED_CHANNEL, and O.AUTHENTIC, and F.MANAGEMENT is derived from O.MANAGEMENT. The SFRs satisfying these objectives are used to satisfy the TOE security functions.

3.7.2 Assurance Measures

The assurance measures of the TOE are stated to satisfy the security assurance requirements. Section B.6.2 specifies the assurance measures of the TOE which are claimed to meet the stated assurance requirements.

3.8 ST Conclusion and Discussion

A Security Target for a secure interface between a payment terminal and a remote POS application has been developed. The contents of the ST complies with the CC specification of STs outlined in CC part 1 [CC104]. EAL3+ has been chosen as the appropriate assurance level.

The ST claims conformance to the POS systems PP. This has been done by moving some of the security objectives and security functional requirements to the TOE IT environment in accordance with the application notes in the PP. A refined model of the device interface model from the PP has been defined in order to describe the TOE and TOE boundaries in relation to the general POS IT environment.

One of the main issues when developing the ST was to identify which security objectives and related security functional requirements to be moved from TOE scope of control to the TOE IT environment scope of control. This was done in order to claim conformance to the PP and a satisfying result of these operations is accomplished. It is therefore concluded that this, at least from a development point of view, is a applicable approach. Additionally, it is believed that the approach most likely have resulted in a more coherent ST than if the ST was developed individually without any PP conformance claims. This is because the requirements for the IT environment are detailed and well argued thereby augmenting the enforcement of the TOE security. The ST development has been easier to approach because the PP focus on POS devices and data flows.

Topics to consider for future versions of the ST include requirements for availability, e.g. in relation to minimum uptime of communication channels and minimum response times during payment transactions. No relevant SFR components are found in CC part 2 addressing these issues of TOE availability and performance, hence explicit security requirements may be stated to achieve this.

CHAPTER 4

Design

In this chapter an example of a hosted POS system design is shown. Afterwards a design example of how a ST compliant secure POS interface for payment solutions may be implemented in this hosted POS system environment is described.

4.1 Design of Hosted POS System Environment

This section describes an example of a hosted POS system. It is to be used as a POS IT environment when designing the TOE security functions as specified in the ST. As the TOE environment must comply with the PP requirements, these have been taken into consideration when designing the system. Because no security target for a complete POS system complying with the PP has been developed, it is not possible to design a system environment fully compliant with the ST. However, the sketched system is an example of how a secure hosted POS system *might* look like in order to make a realistic IT environment for the TOE specified in the ST.

The POS system outlined in this section is based on a POS application which is fully integrated with a standard financial accounting system. A POS system like this could be Navi Partner Retail[1] which is an add-on module to Microsoft Business Solution – Navision[2]. The Navision application includes its own client-server system as well as database management system. The client-server system is, however, not suitable as means of remote access to the POS application from the point of sale. It is designed to be used on (secure) LANs and the application level of client/server communication security is not well documented.

To provide secure access to the POS application for the users at the point of sale, a Citrix MetaFrame[3] solution is introduced. A Citrix MetaFrame Presentation Server provides secure remote access to the POS application via a Citrix Client installed on the cash register PCs in the store. The Citrix Client connection uses TLS [DA99] to secure any input/output data flow it encloses, in a similar way as specified for the Payment Application Data Flows of the TOE. A ST has been developed for the Citrix MetaFrame Presentation Server [Cit04] which describes these security requirements in detail. The input/output device data flows which are able to flow protected via the Citrix Client connection includes data flows from the following input/output devices: printer, display, keyboard and keyboard extensions, e.g bar code scanners and magnetic stripe card readers.

The POS system is illustrated in figure 4.1.

[1] `www.np-retail.dk`

[2] `www.microsoft.com/danmark/mbs/losninger/navision.asp`

[3] `www.citrix.com`

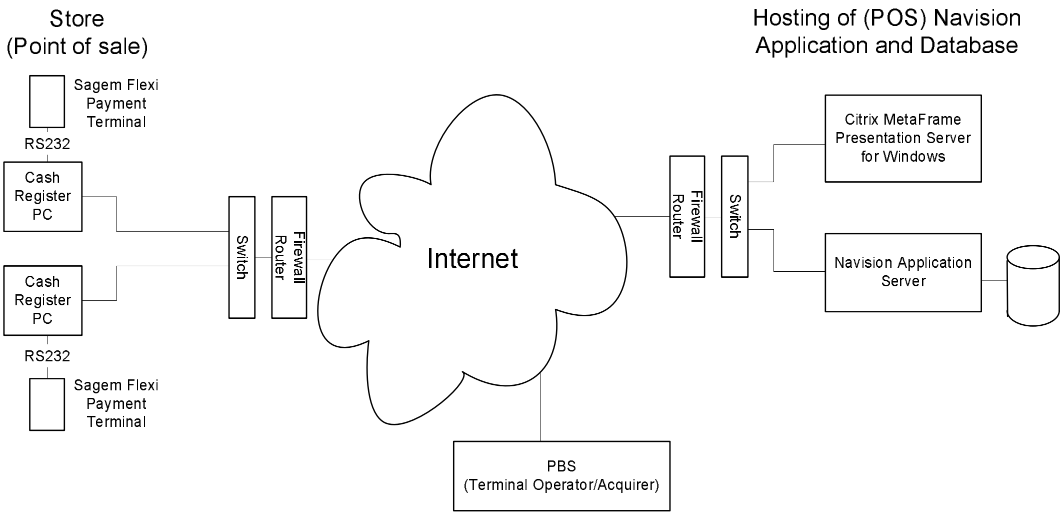
Store
(Point of sale)
Sagem Flexi Payment Terminal
RS232
Cash Register PC
Cash Register PC
RS232
Sagem Flexi Payment Terminal
Switch
Firewall Router
Internet
PBS
(Terminal Operator/Acquirer)
Hosting of (POS) Navision Application and Database
Firewall Router
Switch
Citrix MetaFrame Presentation Server for Windows
Navision Application Server

Figure 4.1: Example of hosted POS system design.

4.1.1 Store

At the point of sale, cash registers are based on a standard PC with the Citrix Client software installed on a Microsoft Windows XP operating system. Each cash register PC may have several input and output devices attached, e.g. keyboards, bar code scanners, receipt printers, and payment terminals. Only the payment terminal is explicitly illustrated in figure 4.1 and is in this example defined as a Sagem Flexi Terminal[4] connected via the serial RS232 interface. In this configuration the payment terminal uses the LAN interface of the PC to establish the connection to the terminal operator and acquirer, which in Denmark is PBS A/S[5].

The figure illustrates two cash registers but it could have been any number depending on the needs for the individual store. Most stores also have a back office PC used by the financial administrator to do the financial accounting and business management. The back office PC is usually not configured for POS operation and does not have all the specialized POS devices, like a payment terminal, attached.

4.1.2 Hosting

At the hosting side of the system two main servers are located. The first is the Citrix MetaFrame Presentation Server which provides the Navision Client Application, i.e. the POS application client, to the Citrix Client. The Citrix Server also implements the overall access control to the POS application. The second server is the Navision Application Server which implements the actual Navision (POS) application and the database management system.

[4] www.sagemdenmark.dk

[5] www.pbs.dk

The Navision Client communication between the Citrix Server and the Navision Server shall be protected against unauthorized access to prevent disclosure and modification. Therefore, it is very important to secure the LAN in the hosting environment, e.g. by installing well configured firewalls and preventing unauthorized physical access. This setup is very scalable and in large hosting environments several Citrix Servers may be needed to host all cash register users and more Navision Servers may be installed to host POS applications if several stores and store chains share hosting environment.

The hosting environment shall be physically protected. Only authorized administrators may be granted access to servers, back-up tapes, and network devices and they shall be protected against fire and theft. The network infrastructure and power supply shall also be very reliable in order to keep availability and uptime of servers at a high level. The main power supply is backed up with battery power supply for short term power supply failures, and on large hosting environments diesel generators will back up the battery supply if the failure lasts longer. Data back-up is performed with a tape station every day, and tape rotation shall be performed manually by an administrator. The back-up tapes shall be stored securely at another location and be protected from theft and fire.

Few POS system providers are able to maintain a hosting environment where physical and infrastructural conditions are at this kind of level. But some companies are specialized in providing these type of facilities in large hosting centers, e.g. TDC Hosting[6] provides very good facilities.

4.1.3 Conformance Claim

As described earlier, the POS IT environment described cannot claim conformance to the ST because no ST for the an entire POS system like this has been developed. But to describe a design example of the ST TOE a realistic POS IT environment must be described first.

When designing the environment, the requirements from the PP has been taken into consideration. An analysis of the input/out device data flows has not been performed but as they are protected in a similar way as specified for the payment terminal data flows described in the ST, the resulting data flow control policy would most likely be fulfilled. The requirements on back-up routines and the physical requirements are also considered in the design. The POS application dependent requirements like session locking, management of TSF and user data, etc are assumed to be implemented in compliance with the PP and an eventual ST specification.

[6] www.tdchosting.dk

4.2 Design of the TOE

The TOE consists of two main components as described in section B.2, the Payment Application (PA) and the Payment Application Client (PAC).

The PA is the server-side of the TOE and is interfacing with the device driver of the payment terminal. It is to be implemented as a "service" (or daemon) application running on the cash register PC. The device driver for the Sagem Flexi Terminal is called the *COM Bridge* and is not really a device driver. It is merely a Microsoft COM (Component Object Model) interface to the *Sagem Payment Solution*, which is a Java application doing the actual interfacing with the payment terminal. Referring to figure B.2 in the ST, the interface between the PA and the COM Bridge which defines the boundaries of the TOE at this end of the system, i.e. the COM Bridge and the Sagem Payment Solution are both parts of the IT environment.

The PAC is the client-side of the TOE and is interfacing with the POS application. As the POS application is fully integrated with the Navision Application, it will be the Navision Application client which implements the interface. The PAC is to be implemented as a Microsoft OCX/ActiveX component which is also a COM-object, and by that making the interface implementable with most Windows based POS applications, including those based on the Navision Application. This interface defines the boundaries of the TOE at the POS application end of the system.

4.2.1 General Functionality

The primary purpose of the TOE is to protect the data flows between the POS application and the payment terminal when they are transported via the Internet, which is considered an insecure path. The data flows shall be protected against disclosure and modification as described in the Input/output data flow control policy stated in section B.2.4 in the ST and by the functional requirements FDP_IFC and FDP_IFF.

Secondly, the TOE shall be manageable by the administrator and any security relevant event shall be auditable.

The PAC establishes a secure connection to the PA whenever the operator needs to initiate a transactional command on the payment terminal and terminates the connection when the transaction is completed. This is also the case when the financial administrator wants to perform administrative transactions, e.g. balancing of totals. When the PAC is not connected the PA is still active as some events may arise even before the operator or financial administrator initiates a transaction. The customer may e.g. initiate a credit card transaction by swiping or inserting the card into the card reader and thereby starting the initial card-PSAM handshaking (exchange of cryptographic certificates and keys, card application selection, PIN entry, etc.). The PA will, however, wait for the operator initiative before starting the actual financial transaction.

4.2.2 Security Functions

This section presents a possible solution to the IT security functions stated in the ST section B.6.1. These are listed below together with a description of each.

F.AUDIT The audit functions assure that audit records are generated for each relevant security event and send to the audit trail. The POS application provides the functionality to store and review the generated records, as these functions are IT environment requirement.

F.CRYPTOGRAPHIC The cryptographic functions assures that a FIPS 140 level 1 compliant cryptographic service provider is used to implement a trusted channel between the payment application and client. Hence, the functions will ensure the confidentiality and integrity of the data flows.

F.MANAGEMENT The management functions assures that any configurable attributes and functions are manageable by the administrator. The management functions also ensure that only an authorized administrator has access to these and that only secure attributes are accepted for the cryptographic functions.

The following sections describes how to satisfy the IT security functions.

4.2.2.1 F.AUDIT

As stated above, the TOE must be able to generate audit records for each security relevant event and send these to the audit trail. Relevant events are initialization of the trusted channel during transaction initialization, any events causing the data flows described in the ST (i.e. the data flow itself), and whenever a management function is performed. Each audit record shall contain time stamp, event type, result/contained data, and identity of user (or role) causing the event.

The audit records are stored in the audit trail of the POS application. But since events may arise while the PAC is not connected to the PA, as described earlier, some may be stored temporarily at the PA until the client reconnects. This could also be the case if the connection is lost during a transaction. These temporary audit records shall be stored in non-volatile memory and for that the Windows event log may be used. Custom log file systems may be implemented, but when using the built-in event log, the implementor will obtain a stable and reliable system where access control on user and application level is already implemented. The size of temporary audit storage is likely to be very small because it is “flushed” whenever the PAC reconnects and no transaction can be carried out before the operator initiative, i.e. the PAC is connected.

4.2.2.2 F.CRYPTOGRAPHIC

The main purpose of the TOE is to provide a trusted channel between the PA and the PAC by means of cryptographic functions. As stated in section B.5 of the ST, cryptographic functions shall be FIPS 140 level 1 compliant. This means that a FIPS 140 level 1 validated Cryptographic Service Provider (CSP) shall be used to implement the trusted channel.

As the TOE is implemented in a Windows environment it is obvious to use a native FIPS 140 level 1 validated CSP. All newer versions of Windows comes with *The Microsoft Enhanced Cryptographic Provider* (RSAENH) described in [Mic05]. Not all algorithms provided by RSAENH are FIPS 140 level 1 validated but the ones complying are clearly marked. This, of course, means that only the validated functions can be used to implement the trusted channel and it is the responsibility of the implementor that this is the case.

As stated in the ST the trusted channel shall be implemented with the *Transport Layer Security* (TLS) protocol defined in [DA99] and [Cho02]. This protocol provides encryption and data integrity between two communicating applications. It is also used to provide mutual authentication of the PA and the PAC.

TLS shall be implemented with one of the following cipher suites in order to comply with the ST:

- *TLS_RSA_WITH_3DES_EDE_CBC_SHA*
- *TLS_RSA_WITH_AES_128_CBC_SHA*
- *TLS_RSA_WITH_AES_256_CBC_SHA*

TLS_RSA_WITH_3DES_EDE_CBC_SHA means that the TLS protocol is implemented with RSA public key encryption used for initial mutual authentication of the PA and PAC and for exchange of symmetric session keys. The certificates used for authentication shall be X.509.v3 certificates [RHS99]. Symmetric cryptography is used for data encryption in form of 3DES EDE (Encrypt-Decrypt-Encrypt) in CBC (Cipher-Block-Chaining) mode. The connection is reliable and includes a message integrity check using a keyed MAC. SHA is used for MAC computations. The two latter cipher suites are identical with the first, except that the symmetric encryption algorithm used is AES applied with key lengths of 128 or 256 bits respectively.

RSA public key encryption will be implemented with 1024-bit keys as this is the key length recommenced by RSA Security Inc.[7] for corporate use.

[7] `www.rsasecurity.com/rsalabs/node.asp?id=2218`

4.2.2.3 F.MANAGEMENT

The TOE shall provide functions to assure that any configurable attribute or security function is manageable by an administrator. The following management functions are to be implemented by the TOE as defined in the SFR component FMT_SMF.1:

a) Functions to manage the audit behavior of the TOE.

b) Functions to manage the cryptographic functions.

The interface to the management functions shall be restricted in a way that permits access for authorized administrators only.

4.3 Design Conclusion and Discussion

A design example compliant with the ST has been developed. The design fulfills the security requirements defined in the ST and is based on standard system components already used to implement actual POS systems. These components include the Windows XP operating system, Citrix MetaFrame, Navision, and standard payment terminals.

Since a design is possible it is concluded that the set of requirements stated in the ST are realistic and usable in terms of TOE implementation.

The design is still very informal and may be considered the initial iteration in order to satisfy the assurance measures defined by the assurance requirements *ADV_FSP.1 — Informal Functional Specification* and *ADV_HLD.2 — Security Enforcing High-Level Design.*

CHAPTER 5

Conclusion and Discussion

5.1 Conclusion

The problem statement defines three objectives to be accomplished when analyzing and defining security requirement of a POS system. These are development of a Protection Profile, a Security Target, and a design example based on the ST.

A Protection Profile for POS systems has been developed. A general model of a POS system has been introduced in order to define the TOE and address any type of POS system. The PP has been specified such that TOEs, which are only a part or component of an entire POS system may also claim conformance to the PP. Under normal circumstances this will not be possible because the CC does not allow partial conformance claims.

A Security Target for a secure interface between a payment terminal and a remote POS application has been developed. The ST has been defined such that it is able to claim conformance to the POS systems PP. This is done by utilizing a refined model of the device interface model from the PP. The device interface model describes the TOE and TOE boundaries in relation to the general POS IT environment.

A design example compliant with the ST has been developed. The design fulfills the security requirements defined in the ST and is based on standard system components already used to implement actual POS systems. Since this is possible it is concluded that the set of requirements stated in the ST are realistic and usable in terms of TOE implementation.

Thereby, all three goals are accomplished with satisfactory results and thereby the aim of the project is fulfilled.

5.2 Discussion of the CC

The Common Criteria is difficult to get familiarized with as it is comprehensive and literature on how to approach it is sparse. However, many Protection Profiles and Security Targets are available and these may form a good foundation of inspiration. Once familiarized with the organization of the CC it is realized that it is logically structured.

The great advantage of the CC is that security functional requirements and security assurance requirements are stated independently of each other. This makes it possible to state the evaluation assurance level desired, regardless of the comprehensiveness of the actual security functional requirements. However, if used inappropriately this is also the greatest disadvantage of the CC. This is due to the fact that a high evaluation assurance level may be claimed to a product with relatively weak security requirements. This might lead the consumer to think that the product provides higher security than it actually does because consumers normally compare products by the EAL stated.

In order to counter this dilemma, widely accepted Protection Profiles must be developed for more and more product areas, e.g. POS systems. All products in one area must claim conformance to the same relevant PP. This will make the products comparable in terms of minimum security functional requirements.

5.3 Perspective

The obvious next step is to have the PP and ST evaluated by one of the official CC evaluation labs. The evaluation will most likely result in an iterative process where changes suggested by the evaluator are implemented until an evaluated product is achieved.

The Common Criteria version 3.0 is currently reviewed in public and is expected to be approved for release in July 2006. According to the official Common Criteria web site[1] the new version supports composition of compatible certified products, i.e. most likely countering one of the main challenges in developing the POS systems PP. Hence, it may be relevant to upgrade or rewrite the PP to comply with the new version. Additionally, version 3.0 has been greatly simplified by rewriting and drastically reducing the number of classes, families, and components in part 2. This will make it easier to approach.

[1] www.commoncriteriaportal.org

Bibliography

[APA03] APACS. *APACS PIN Entry Device Protection Profile ver. 1.37*, July 2003. Available from www.commoncriteriaportal.org.

[CC104] Common Criteria. *Common Criteria for Information Technology Security Evaluation ver. 2.2 - Part 1: Introduction and general model*, January 2004. Available from www.commoncriteriaportal.org.

[CC204] Common Criteria. *Common Criteria for Information Technology Security Evaluation ver. 2.2 - Part 2: Security functional requirements*, January 2004. Available from www.commoncriteriaportal.org.

[CC304] Common Criteria. *Common Criteria for Information Technology Security Evaluation ver. 2.2 - Part 3: Security Assurance Requirements*, January 2004. Available from www.commoncriteriaportal.org.

[Cho02] P. Chown. *Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS).* Skygate Technology, June 2002.
Available from www.faqs.org/rfcs/rfc3268.html.

[Cit04] Citrix Systems Inc. *Security Target for Citrix MetaFrame XP Presentation Server For Windows with Feature Release 3 ver 1.6*, March 2004. Available from www.commoncriteriaportal.org.

[DA99] T. Dierks and C. Allen. *The TLS Protocol.* The Internet Society, January 1999.
Available from http://www.faqs.org/rfcs/rfc2246.html.

[DSLV02] Christian Stuble Dr. Steffen Lange, Dr. Andreas Nonnengart and Roland Vogt. *Discretionary Information Flow Control Protection Profile ver. 2.01.* DFK, September 2002. Available from www.commoncriteriaportal.org.

[Eur01] Europay International, PBS A/S and Visa International Service Association. *Terminal Architecture for PSAM Applications (TAPA) - Application Architecture Specification, version 2,1*, February 2001. Available from `http://international.visa.com/fb/downloads`.

[Inf99] Information Systems Security Organization. *Controlled Access Protection Profile*, 1.d edition, October 1999.
Available from `www.commoncriteriaportal.org`.

[Mic02] Microsoft. *Windows 2000 Security Target ver. 2.0*, October 2002. Available from `www.commoncriteriaportal.org`.

[Mic05] Microsoft. *Microsoft Enhanced Cryptographic Provider*, February 2005. Available from `www.msdn.com`.

[Ora98] Oracle. *Commercial Database Management System Protection Profile ver. 1.0*, March 1998. Available from `www.commoncriteriaportal.org`.

[PBS04] PBS A/S. *Technical Reference Guide - Open Terminal Requirement Specification (OTRS), version 2.4*, March 2004.
Available from `www.forretning.pbs.dk/materiale/termspec.htm`.

[RHS99] W. Polk R. Housley, W. Ford and D. Solo. *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, January 1999. Available from `www.faqs.org/rfcs/rfc2459.html`.

[U.S00] U.S. Department of Commerce and National Institute of Standards and Technology. *FIPS 186-2 Digital Signature Standard (DSS)*, January 2000.
Available from `http://csrc.nist.gov/publications/fips/`.

[Whe05] Lynn Wheeler. *Security Glossary*, 2005.
Available from `www.garlic.com/~lynn/secgloss.htm`.

APPENDIX A

Protection Profile

A.1 PP Introduction

A.1.1 PP Identification

Title: Point of Sale System CC Protection Profile

Authors: Anders Hedegaard and Allan Pedersen

Publishing Date: 1st August 2005

PP Version Number: 1.0

Version of CC used for development: CC version 2.2

A.1.2 PP Overview

A Point of Sale system is defined as an IT system used to register sales and payments at the Point of Sale into an audit trail. The Point of Sale System CC Protection Profile, hereafter called the POSPP, has been developed to specify the security functional and assurance requirements needed to protect a Point of Sale system.

The POSPP is written to cover all kinds of POS systems as it is described in generalized terms.

A.1.3 PP Organization

The organization of this PP is based on appendix B of part 1 of the Common Criteria for Information Technology Security Evaluation (CC) [CC104]. Instead of collecting all application notes in a separate section all application notes relevant for a specific section or paragraph will appear immediately after the indicated section or paragraph.

This document is not meant to be self-contained. Whenever necessary the CC documentation should be consulted for additional information and guidance, e.g. in conjunction with reading the Security Function Requirements and Security Assurance Requirements.

A.1.4 Abbreviations

The following abbreviations are used throughout the POSPP.

CC Common Criteria

EAL Evaluation Assurance Level

IT Information Technology

OSP Organizational Security Policy

PIN Personal Identification Number

POS Point Of Sale

POSPP Point Of Sale CC Protection Profile

PP Protection Profile

SFP Security Function Policy

SFR Security Function Requirement

SOF Strength Of Function

ST Security Target

TOE Target Of Evaluation

TSF TOE Security Function

TSP TOE Security Policy

A.2 TOE Description

Point of sale (POS) systems for handling payments are widely deployed in commercial outlets of all types, from petrol stations and kiosks to department stores and marts. Typically, at the actual point of sale there are one or more PCs, to each of which is attached a cash register, printer(s), bar code scanner and a payment terminal for credit card transactions.

A.2.1 Point of Sale System

A Point of Sale is the physical location at which goods are sold to customers. Point of sale (POS) systems are IT systems designed to register sales and payments at the point of sale into an audit trail. The audit trail is in general used to store evidence of financial transactions and any auditable security relevant event. The audit trail is used to store both financial and security audit records. The POS system shall be able to produce evidence of registered sales and payments from the audit trail.

The audit trail shall be securely stored and handled in accordance with applicable laws as this is the actual evidence of the individual sale registration. Furthermore, legislation may dictate the quality of and what data to register in the audit trail.

A.2.2 Roles

Roles included in this Protection Profile are *Customer*, *Operator*, *Financial Manager*, and *Administrator*:

Customer Normally, the customer is assumed not to be authorized and interacts only with the TOE via the operator. Only in specialized operations of the POS system the customer may be asked to interact directly with the TOE, e.g. during a transaction via a payment terminal attached to the POS system, where the customer can be asked to swipe a credit card and enter a PIN.

Operator The operator is an authorized user of the TOE who is responsible for the actual registration and handling of the sold goods and received payments at the point of sale, e.g. a sales clerk. Furthermore, the operator is responsible for providing the produced evidence of the sale to the customer.

Financial Manager The financial manager is an authorized user of the TOE who is able to pull out information from the audit trail in relation to accounting.

Administrator The administrator is an authorized user of the TOE who is responsible for installation, configuration and maintenance of all functions of the POS system.

More roles may be identified and a more detailed division of the specified roles may be argued. TOEs claiming conformance to this PP shall at least maintain the mentioned roles. The roles are used to assign users with access rights to data and to define the *data flow control policies*, see A.2.3.1 for details.

A.2.3 Data Flows in the POS System

The data flows in the system are traffic of data in and out of the audit trail. The incoming data flows may arise from different input devices attached to the TOE such as bar code scanners, keyboards, payment terminals, etc. Typically, incoming traffic is caused by registration of sales and payments. Outgoing data flows are normally used to produce evidence of registered sales and payments. Evidence may be printed receipts and invoices, or a customer display showing the last sale or payment registered.

One input/output device may be the source or destination of more than one data flow with different demands to the *data flow control policy.* This could be a printer used to print receipts to customers, which is one data flow, and financial reports destined for the financial manager only, which is another data flow. Figure A.1 illustrates the data flows in the POS system.

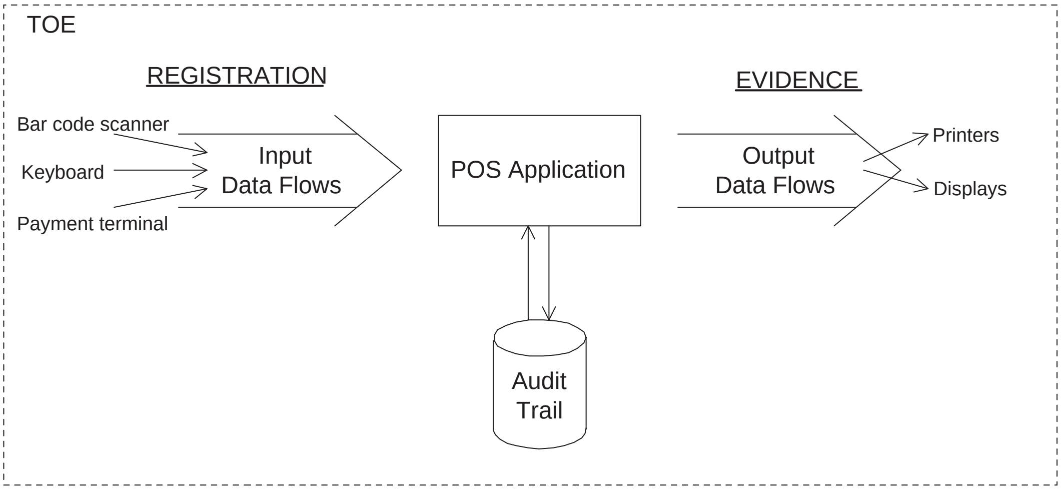

Figure A.1: TOE data flows.

In addition, figure A.1 introduces the POS application. In practice, the data flows will not flow directly from the input devices into the audit trail and directly from the audit trail to the output devices. Usually, the data will undergo different processing in the POS application before actually being stored in the audit trail. The processing could be data validation, price lookup, print generation etc.

To make creation of data flows between the attached devices and the audit trail through the POS application possible, each input/output device must have a well defined interface to the POS application. Figure A.2 illustrates this. In the illustration the device stub of the interface is generalized as a device driver, although it may be implemented by other means. Normally, a device will have its own interface but some related devices may share a common interface and device driver. For instance, some bar code scanners are implemented as keyboard extensions, hence they will share the "standard input" interface of a PC.

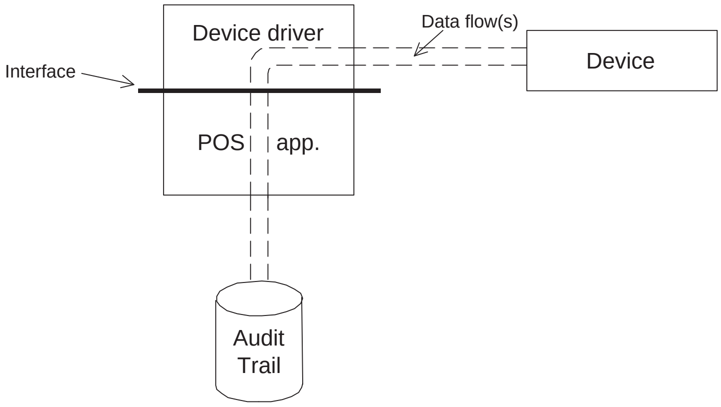

Figure A.2: Interface between attached device and POS application.

A.2.3.1 Data Flow Control Policies

As stated in section A.2.1, the audit trail must be stored and handled securely. For this to make sense, it is equally important to handle the data flows in and out of the audit trail securely as well. The system shall ensure that the data flowing from input devices into the audit trail is not manipulated or in any other way compromised and likewise for the data flowing from the audit trail to output devices.

To ensure required data security for the data flows a *data flow control policy* shall be made individually for each identified data flow. And to determine how to protect the individual data flows a threat analysis must be carried out for each of these. The threat analysis shall identify the probability and consequences of an attacker compromising the data flow. Attributes to consider includes:

- Type of input/output device used in the data flow.
- Role of user creating and receiving the data.
- Type and sensitivity of the data.
- Media in which the data flows.
- Possible threat agents.

Data flows may be grouped if they have similar security attributes and hence will have equal data flow control policies. When the analysis is conducted it is possible to determine which countermeasures, if any, are necessary to achieve the desired level of protection. The level of protection is divided into three categories:

- **Low** Minimum standard countermeasures are required to achieve desired data security.
- **Medium** Additional countermeasures above the minimum level of protection are required.
- **High** Most stringent protection and rigorous security countermeasures are required.

It is up to the ST author to determine exactly which countermeasures are categorized under *Low*, *Medium*, and *High* level of protection as it is highly implementation dependent.

A.3 TOE Security Environment

A.3.1 Assumptions

This section describes the assumptions made for the TOE environment and intended method of use of the TOE.

A.NO_EVIL It is assumed that administrators of the TOE are competent of managing and maintaining the TOE and the security of the functions and data it contains. It is also assumed that administrators do not have evil intentions of abusing their privileges.

A.3.2 Threats

This section describes TOE assets and threats to TOE.

A.3.2.1 Assets

The primary asset to protect is the audit trail. If the audit trail is lost or maliciously manipulated the evidence of the registered sales and payments cannot be restored. This may lead to incomplete financial accounting which may conflict with legislation. In addition, the audit trail contains valuable information to the owner of the POS system, e.g. sales statistics, which is also valuable to attackers in relation to industrial espionage.

Secondary assets to protect are security attributes of the TOE security functions (TSF) such as user names and passwords, cryptographic keys, etc.

A.3.2.2 Threat Agents

Threat agents are categorized as *authorized users* and *unauthorized users*. Note that administrators are not considered threat agents due to the assumption **A.NO_EVIL**. In the following all threat agents are referred to as *attackers*.

Attackers are assumed to have various levels of expertise, motivation, and resources available. The expertise may come from specialized knowledge of the TOE. Motivation will normally arise from economic gain but also personal revenge may be a motivation.

A.3.2.3 Threats

The following threats are identified:

T.ACCESS An attacker may try to gain unauthorized access to the information protected by the TOE. This could be an unauthorized user impersonating an authorized user, or it may be an authorized user impersonating a, perhaps, more privileged user.

T.MODIFICATION An attacker may try to modify information protected by the TOE maliciously.

T.PHYSICAL The audit trail may physically be lost due to fire, theft, force majeure, etc.

T.UNATTENDED_SESSION An attacker may gain unauthorized access to the TOE via a unattended session.

T.INCOMPETENCE A user may compromise the security of the TOE due to incompetent usage.

T.DATA_FLOW An attacker may compromise the integrity of an input/output data flow.

A.3.3 Organizational Security Policies

This section states the organizational security policies (OSPs) for the TOE.

P.AUTHORIZED_USERS Only authorized users may access the TOE.

P.ACCOUNTABILITY Authorized users of the TOE shall be held accountable for their actions within the TOE.

P.TRAIN Authorized users accessing functions of the TOE shall receive continuous training in secure use of the TOE.

A.4 Security Objectives

A.4.1 Security Objectives for the TOE

The following objectives states the security objectives of the TOE:

O.IA The TOE shall provide means for identifying and authenticating users before allowing access to the TOE and its resources.

O.MANAGE The TOE shall provide functionality which enables authorized administrators to manage and support the security attributes of the TOE, and restrict these functions from unauthorized use.

O.AUDIT The TOE shall provide functionality to record security relevant events in sufficient detail to help administrators of the TOE to hold individual users accountable for any actions they perform that are relevant to the security of the TOE.

O.DATA_FLOW For attached input/output devices a data flow control policy based on a threat analysis shall be made for each identified data flow. This is done to accommodate the different demands to secure communication of the devices.

O.SESSION A session shall only be active when an authorized user is interacting with the TOE interface. Therefore, the TOE shall provide functionality for the user to lock the current interactive session. It should also be possible for the TOE to automatically lock the session if the user is considered inactive. The user must re-authenticate to unlock the session. Furthermore, the user should re-authenticate before each sale and/or payment transaction.

O.BACK-UP The TOE shall provide functionality for administrators to back up the data in the system in order to make it possible to restore, as a minimum, the audit trail in case of hacking, hardware failure, fire, theft, force majeure, etc.

A.4.2 Security Objectives for the Environment

In the general definition of a POS system the TOE is assumed to be self-contained, hence there are no dependencies on an IT-environment. But for the general non-IT environment the following objectives shall be met:

OE.TRAIN The overall responsible for the TOE shall arrange training for all authorized users of the TOE including the administrators.

OE.ADMIN_VETTING The overall responsible for the TOE shall perform vetting of administrators to ensure that they are competent and non-hostile.

OE.PHYSICAL The TOE shall be physically protected in such a way that attackers cannot remove the TOE or parts of the TOE which are critical to the security of the TOE, or in other ways physically compromise the TOE and the data it contains, i.e. the audit trail, security attributes, etc.

A.5 IT Security Requirements

A.5.1 TOE Security Functional Requirements

This section describes the security functional requirements (SFR) components that must be satisfied by the TOE for claiming conformance with this protection profile. The components are taken from CC part 2 [CC204] and in table A.1 all identified SFRs are listed for a quick overview.

Class	**Family**	**Component**
FAU	FAU_GEN	FAU_GEN.1
		FAU_GEN.2
	FAU_SAR	FAU_SAR.1
		FAU_SAR.2
	FAU_STG	FAU_STG.1
FDP	FDP_IFC	FDP_IFC.1
	FDP_IFF	FDP_IFF.1
FIA	FIA_UAU	FIA_UAU.2
		FIA_UAU.6
	FIA_UID	FIA_UID.2
FMT	FMT_MOF	FMT_MOF.1
	FMT_MSA	FMT_MSA.1
		FMT_MSA.3
	FMT_MTD	FMT_MTD.1
	FMT_SMF	FMT_SMF.1
	FMT_SMR	FMT_SMR.1
FPT	FPT_STM	FPT_STM.1
FTA	FTA_SSL	FTA_SSL.1
		FTA_SSL.2

Table A.1: SFR components.

In the following the TOE security functional requirements are listed in detail. The TSFs are listed in the same order as in the catalog. Text in *italic* indicates that an assignment or refinement has been performed.

A.5.1.1 FAU — Security Audit

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:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the [selection: choose one of: minimum, basic, detailed, not specified] level of audit; and

c) [assignment: other specifically defined auditable events].

Application note: *It is left to the ST author to assign the level of audit and other specifically defined auditable event if suitable.*

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: other audit relevant information]

Application note: *It is left to the ST author to assign other audit relevant information to be included in the audit records*

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_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide *the administrators* with the capability to read *any 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 from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to *prevent* unauthorized modifications to the audit records in the audit trail.

A.5.1.2 FDP — User Data Protection

FDP_IFC.1 Subset information flow control

FDP_IFC.1.1 The TSF shall enforce the *[assignment: input/output device data flow control SFP]* on *[assignment: input/output devices which acts as TOE information interfaces and causes information to flow into and out of the audit trail.]*

Application note: *FDP_IFC.1 is to be iterated for each identified input/output device data flow. Several iterations may be necessary for each input/output device. An iteration of FDP_IFF.1 shall be made for each iteration of FDP_IFC.1 respectively.*

FDP_IFF.1 Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the *[assignment: input/output device data flow control SFP]* based on the following types of subject and information security attributes: *[Assignment: list of security attributes to be used to conduct a threat analysis of the data flow.]*

Application note: *The threat analysis is conducted to determine the level of protection of the relevant data flow (low, medium, or high). See section A.2.3 for more details. It is left to the ST author to list the security attributes and perform the threat analysis.*

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:

a) A threat analysis of the [Assigment: input/output device data flow] is carried out.

b) and the following countermeasures to achieve desired [selection: low, medium, high] level of protection for the data flow are implemented: [assignment: list of identified necessary countermeasures]

Application note: *As it is highly implementation dependent, it is left to the ST author to define which countermeasures are necessary to achieve the desired level of protection for the data flow. See section A.2.3 for more details. The ST author shall also select which level of protection is concluded to be necessary for the data flow.*

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 authorize an information flow based on the following rules: [assignment: rules, based on security attributes, that explicitly authorize 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 authorize information flows].

A.5.1.3 FIA — Identification and Authentication

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.6 Re-authenticating

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions:

a) The session has been locked or terminated.

b) A new sales and/or payment transaction is to be initiated.

c) [assignment: additional conditions under which re-authentication is required].

Application note: *The conditions under which a session is locked or terminated are defined in FTA_SSL. It is left to the ST author to assign additional conditions for which re-authentication is required. This could be during special actions, e.g. saving security attributes, recovering data from backups or handling extraordinary sales or payments.*

FIA_UID.2 User identification before any action

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF mediated actions on behalf of that user.

A.5.1.4 FMT — Security Management

FMT_MOF.1 Management of security functions behavior

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

a) The functions implementing the security auditing, including which security events to audit.

b) The functions implementing the input/output device data flow control policies for the attached input and output devices.

c) The functions implementing the method of identification and authorization of users.

d) The functions implementing timers and the clock synchronization.

e) The functions implementing the system backup routines.

f) The functions implementing the session locking methods.

g) [Assignment: additional manageable functions].

to *the administrators.*

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

FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the *[Assigment: input/output device data flow control SFP]* to restrict the ability to *modify, [assignment: other operations]* the security attributes *referenced in the indicated policy* to *the administrators.*

FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the *Assigment[input/output device data flow control SFP]* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the *administrators* 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

a) The security audit trail.

b) The TOE system clock.

c) [Assignment: additional TSF data]

to *the administrators*

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 security management functions:

a) Functions to assign and maintain lists of users and roles.

b) Functions to create and recover backups of, as a minimum, the audit trail.

c) Functions to set up and manage information flow controls for input and output devices.

d) Functions to manage the TOE system clock and timers.

e) Functions to manage and review the security audit trail.

f) Functions to manage session locking attributes.

g) [Assignment: other security management functions]

Application note: *It is left to the ST author to specify additional implementation dependent security management functions*

FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles:

a) Customer

b) Operator

c) Financial Manager

d) Administrator

e) [assignment: other identified roles].

Application note: *It is left to the ST author to assign other identified roles.*

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

Application note: *It may be argued that the Customer role normally does not need to have users assigned because they act as "anonymous" users identified and authorized by the Operator. See section A.2.2.*

A.5.1.5 FPT — Protection of the TSF

FPT_STM.1 Reliable time stamps

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

A.5.1.6 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:

a) clearing or overwriting display devices, making the current contents unreadable;

b) 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: *Re-authorization of user.*

Application note: *It is left to the ST author to define the time interval of user inactivity. It is to be defined with respect to the general physical accessibility of the TOE and how the user-initiated locking TSF is implemented (FTA_SSL.2).*

FTA_SSL.2 User-initiated locking

FTA_SSL.2.1 The TSF shall allow user-initiated locking of the user´s own interactive session, by:

a) clearing or overwriting display devices, making the current contents unreadable;

b) disabling any activity of the user´s data access/display devices other than unlocking the session.

FTA_SSL.2.2 The TSF shall require the following events to occur prior to unlocking the session: *Re-authorization of user.*

Application note: *The time interval of user inactivity of the TSF-initiated session locking is to be defined with respect to the implementation of the user-initiated locking. If the locking is implemented in a way in such a way that the user cannot leave the TOE physically without locking a possible interactive session, e.g. with a smart-card attached to a key-chain, the time interval may be very long. It may even be infinite if it can be argued unnecessary due to stringent user-initiated locking functionality.*

A.5.2 TOE Security Assurance Requirements

This section describes the security assurance requirement. The assurance level is given by EAL3 with no augmentation from CC part 3 [CC304].

Class	Component	
ACM	ACM_CAP.3	Authorization controls
	ACM_SCP.1	TOE CM coverage
ADO	ADO_DEL.1	Delivery procedures
	ADO_IGS.1	Installation, generation and start-up procedures
ADV	ADV_FSP.1	Informal functional specification
	ADV_HLD.2	Security enforcing high-level design
	ADV_RCR.1	Informal correspondence demonstration
AGD	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
ALC	ALC_DVS.1	Identification of security measures
ATE	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: high-level design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
AVA	AVA_MSU.1	Examination of guidance
	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.1	Developer vulnerability analysis

Table A.2: Security assurance components.

A.5.2.1 ACM — Configuration Management

ACM_CAP.3 Authorization controls

Developer action elements

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

ACM_CAP.3.2D The developer shall use a CM system.

ACM_CAP.3.3D The developer shall provide CM documentation.

Content and presentation of evidence elements

ACM_CAP.3.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.3.2C The TOE shall be labelled with its reference.

ACM_CAP.3.3C The CM documentation shall include a configuration list and a CM plan.

ACM_CAP.3.4C The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.3.5C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.3.6C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.3.7C The CM system shall uniquely identify all configuration items.

ACM_CAP.3.8C The CM plan shall describe how the CM system is used.

ACM_CAP.3.9C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.3.10C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.3.11C The CM system shall provide measures such that only authorized changes are made to the configuration items.

Evaluator action elements

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

ACM_SCP.1 TOE CM coverage

Developer action elements

ACM_SCP.1.1D The developer shall provide a list of configuration items for the TOE.

Content and presentation of evidence elements

ACM_SCP.1.1C The list of configuration items shall include the following: implementation representation and the evaluation evidence required by the assurance components in the ST.

Evaluator action elements

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

A.5.2.2 ADO — Delivery and Operation

ADO_DEL.1 Delivery procedures

Developer action elements

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

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

Content and presentation of evidence elements

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

Evaluator action elements

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

ADO_IGS.1 Installation, Generation, and Start-up Procedures

Developer action elements

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

Content and presentation of evidence elements

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

Evaluator action elements

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

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

A.5.2.3 Development

ADV_FSP.1 Informal functional specification

Developer action elements

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

Content and presentation of evidence elements

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

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

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

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

Evaluator action elements

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

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

ADV_HLD.2 Security Enforcing High-level Design

Developer action elements

ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

Content and presentation of evidence elements

ADV_HLD.2.1C The presentation of the high-level design shall be informal.

ADV_HLD.2.2C The high-level design shall be internally consistent.

ADV_HLD.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_HLD.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

Evaluator action elements

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

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

ADV_RCR.1 Informal correspondence demonstration

Developer action elements

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

Content and presentation of evidence elements

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Evaluator action elements

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

A.5.2.4 AGD — Guidance Documents

AGD_ADM.1 Administrator guidance

Developer action elements

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

Content and presentation of evidence elements

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

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

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

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

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

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

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

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements

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

AGD_USR.1 User Guidance

Developer action elements

AGD_USR.1.1D The developer shall provide user guidance.

Content and presentation of evidence elements

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

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

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

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

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements

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

A.5.2.5 ALC — Life Cycle Support

ALC_DVS.1 Identification of security measures

Developer action elements

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

Content and presentation of evidence elements

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

ALC_DVS.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

Evaluator action elements

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

ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

A.5.2.6 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 of evidence elements

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

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

Evaluator action elements

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

ATE_DPT.1 Testing: High-level Design

Developer action elements

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

Content and presentation of evidence elements

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

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 of evidence elements

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Evaluator action elements

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

ATE_IND.2 Independent Testing - Sample

Developer action elements

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

Content and presentation of evidence elements

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

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements

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

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

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

A.5.2.7 AVA — Vulnerability Assessment

AVA_MSU.1 Examination of guidance

Developer action elements

AVA_MSU.1.1D The developer shall provide guidance documentation.

Content and presentation of evidence elements

AVA_MSU.1.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.1.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.1.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.1.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

Evaluator action elements

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

AVA_MSU.1.2E The evaluator shall repeat all configuration and installation procedures to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.1.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_SOF.1 Strength of TOE Security Function Evaluation

Developer action elements

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

Content and presentation of evidence elements

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

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

Evaluator action elements

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

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

AVA_VLA.1 Developer vulnerability analysis

Developer action elements

AVA_VLA.1.1D The developer shall perform a vulnerability analysis.

AVA_VLA.1.2D The developer shall provide vulnerability analysis documentation.

Content and presentation of evidence elements

AVA_VLA.1.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2C The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.

AVA_VLA.1.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

Evaluator action elements

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

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

A.6 Application Notes

POS systems are found in many different variants. Everything from a single cash register to large complex systems with many payments terminals, printers, bar code scanners, etc. are seen. This means that a POS system usually is build from many different parts delivered by different manufactures. Therefore, it is necessary to have the possibility to claim conformance with this PP even though only a part of the POS system is to be evaluated, e.g. an input/output device. In this case it is left to the ST author to define the new boundaries of the TOE and move the remaining requirements to the IT environment. At the TOE boundaries an interface must be defined to the rest of the POS system and in the example of the TOE being a input/output device the interface will naturally be the one illustrated in figure A.2.

It is left to the ST author to implement the relevant TOE security functions. The possibility exists that more TSFs than the PP suggest may be implemented. For instance, there are no demands that cryptographic functions shall be implemented but if a threat analysis of the data flow concludes that a high protection level is needed, the class FCS (Cryptographic support) might be added.

A.7 Rationale

A.7.1 Security Objective Rationale

	O.IA	O.MANAGE	O.AUDIT	O.DATA_FLOW	O.SESSION	O.BACK-UP	OE.TRAIN	OE.ADMIN_VETTING	OE.PHYSICAL
A.NO_EVIL							x	x	
T.ACCESS	x	x	x						
T.MODIFICATION	x	x	x			x			
T.PHYSICAL						x			x
T.UNATTENDED_SESSION					x		x		
T.INCOMPETENCE							x		
T.DATA_FLOW		x		x					
P.AUTHORIZED_USERS	x	x							
P.ACCOUNTABILITY	x	x	x		x				
P.TRAIN							x		

Table A.3: Relations illustrating the security objective rationale.

A.7.1.1 Security Objectives Suitable to Uphold Assumptions

The following rationale demonstrates how the objectives cover the assumptions:

A.NO_EVIL OE.TRAIN upholds this assumption because it assures that the administrators stay competent via continuous training.

OE.ADMIN_VETTING upholds this assumption because it ensures that all administrators are vetted to ensure that they stay competent and non-hostile.

A.7.1.2 Security Objectives Suitable to Counter the Threats

The following rationale demonstrates how the objectives counter the threats:

T.ACCESS This threat is mainly countered by O.IA which provides the means to identify and authenticate users before they are granted access to the TOE. In this way services of the TOE are only available if a user is identified which reduces the threat of unauthorized access to the TOE.

O.MANAGE restricts the use of TOE security functions from unauthorized use. This helps to reduce the threat of unauthorized access to the TOE.

O.AUDIT provide the means to record security relevant actions in the TOE. In this way administrators are able to keep track of user actions in the TOE and they can take necessary actions if suspicious activities are recorded.

T.MODIFICATION This threat is mainly countered by O.IA which provides the means to identify and authenticate users before they are granted access to the resources of the TOE, and O.MANAGE which makes the administrators capable of restricting unauthorized use of security related functions.

O.AUDIT provides means to record security relevant actions, e.g new entries in the audit trail or modification of security attributes, in the TOE. In this way unauthorized modification of data in the TOE is tracked and it is possible to restore previous conditions. If more critical modifications have maliciously been performed a backup recovery operation may be required. This is addressed by OE.BACK-UP but should only be used when no other options are available.

T.PHYSICAL This threat is mainly countered by O.BACK-UP which states that administrators shall back up, as a minimum, the audit trail in order to make it possible to restore the data in case of physical loss in case of fire, theft, force majeure, etc. This objective effectively decreases the threat of physical loss, especially if a good back up plan is made.

OE.PHYSICAL states that the TOE shall be physically protected in a way that attackers cannot remove the TOE or in any other way physically compromise it or the data it contains. By securing the TOE physically, e.g. bolting it to the ground, the threat of physical loss is reduced.

T.UNATTENDED_SESSION This threat is mainly countered by O.SESSION which ensures that a session cannot be left unattended. It makes it possible for an authorized user to lock the the current session. If the user leaves the session without locking it the TOE automatically locks the session.

OE.TRAIN ensures that all authorized users of the TOE receives continuously training in use of the TOE. Education reduces the risk of users leaving an open session and thereby leaving the TOE open for attackers.

T.INCOMPETENCE This threat is countered by OE.TRAIN which ensures that all authorized users are continuously trained and educated in secure use of the TOE. This will effectively reduce the threat of users using the TOE in an incompetent way which compromises the security of the TOE.

T.DATA_FLOW This threat is mainly countered by O.DATA_FLOW which ensures that a threat analysis is carried out for each data flow and suitable security measures are implemented.

O.MANAGE ensures that administrators are able to manage TOE security functions in a secure way. In this way the data integrity is preserved and the risk of attacks on the data flows is minimized.

A.7.1.3 Security Objectives Suitable to Meet OSPs

The following rationale demonstrates how the objectives achieve the OSPs:

P.AUTHORIZED_USERS O.IA ensures that the TOE supports authentication and identification of users before they gain access to the TOE. In this manner only authorized users are able to access the TOE.

O.MANAGE ensures that security functions are managed in a way in such a way that only authorized users have access to these.

P.ACCOUNTABILITY O.IA, O.SESSION, and O.AUDIT ensure that users are held responsible for their actions in the TOE. This is because users have to (re-)authenticate and identify themselves before the TSFs allow any action in the TOE. Furthermore, all security relevant actions are recorded which enables administrators to monitor any unusual traffic.

O.MANAGE ensures that security functions are managed in a way which assures that all security relevant actions are recorded.

P.TRAIN OE.TRAIN ensures that authorized users of the TOE receive continuous training in secure use of the TOE.

A.7.2 Security Requirements Rationale

Table A.4 provides the correlation between the security objectives to be met by the TOE.

O.IA The components FIA_UAU.2, FIA_UID.2, and FMT_SMR.1 ensure that users (roles) have to identify and authenticate themselves before any action in the TOE is allowed by the TSF. On special occasions it may be necessary to re-authenticate a user, e.g. on session time-outs. Re-authentication is ensured by FIA_UAU.6.

FMT_SMF.1 provides the security functions to manage the attributes.

O.MANAGE The components FMT_MSA.1, FMT_MOF.1, FMT_MTD.1, and FMT_SMR.1 ensure that only authorized users (roles) are able to manage the

Security Objective	Security Functional Requirement
O.IA	FIA_UAU.2 FIA_UAU.6 FIA_UID.2 FMT_SMF.1 FMT_SMR.1
O.MANAGE	FMT_MOF.1 FMT_MSA.1 FMT_MSA.3 FMT_MTD.1 FMT_SMF.1 FMT_SMR.1
O.AUDIT	FAU_GEN.1 FAU_GEN.2 FAU_SAR.1 FAU_SAR.2 FAU_STG.1 FMT_UID.2 FMT_SMF.1 FMT_SMR.1 FPT_STM.1
O.DATA_FLOW	FDP_IFC.1 FDP_IFF.1
O.SESSION	FIA_UAU.6 FTA_SSL.1 FTA_SSL.2
O.BACK-UP	FMT_SMF.1

Table A.4: Security requirements rationale.

security attributes. FMT_SMF.1 provides the security functions to manage the attributes.

FMT_MSA.3 ensures that the TSF provides default values for relevant security attributes.

O.AUDIT The component FAU_GEN.1 ensures that auditable events are identified for which audit records should be generated and which information the records contain, e.g. user log-out. FAU_GEN.2 associates users with each record.

The components FAU_SAR.1 and FAU_SAR.2 ensures that only users that have been granted explicit read-access are able to read audit records.

FAU_STG.1 ensures that the TSF shall protect the stored audit records from unauthorized deletion. Furthermore the TSF shall prevent unauthorized modifications to the audit records in the audit trail.

The components FMT_UID.2 and FMT_SMR.1 ensures that users (roles) are identified and authenticated before they can interact with the TOE. This makes the previously mentioned user association possible.

FMT_SMF.1 provides the security functions to manage the attributes.

The component FPT_STM.1 ensures that TSFs provide reliable time stamps for its own use. This is necessary to make sure that audit records in the audit trail are reliable.

O.DATA_FLOW The component FDP_IFC.1 ensures that an information flow control SFP is made for each identified data flow. FDP_IFF.1 ensures that the data flow is protected at a level determined by a threat analysis. Furthermore, a data flow is only allowed if a threat analysis of the data flow is carried out and countermeasures to achieve the desired level of protection for the data flow are implemented.

O.SESSION The components FIA_UAU.6, FTA_SSL.1, and FTA_SSL.2 ensures that both TSF- and user-initiated session locking are possible and it requires re-authentication to unlock the session.

O.BACK-UP FMT_SMF.1 provides the security functions to manage the back-up attributes.

A.7.2.1 Dependencies of Security Requirements

Table A.5 gives all dependencies met by the SFRs and thereby proofs that the SFRs are mutually supportive and internally consistent.

A.7.3 Assurance Requirements Rationale

This protection profile has been developed for an environment with moderate level of risk and it is therefore concluded that the level of assurance provided by EAL3 with no augmentation is appropriate. EAL3 gives a moderate level of independently assured security. Compared to EAL2, EAL3 gives more confidence in the fact that the TOE will not be tampered with during development.

SFR	**Dependency**	**Note**
FAU_GEN.1	FPT_STM.1	
FAU_GEN.2	FAU_GEN.1 FIA_UID.2	FAU_GEN.2 has dependency on FIA_UID.1. Since FIA_UID.2 is hierarchical to that component the dependency is fulfilled.
FAU_SAR.1	FAU_GEN.1	
FAU_SAR.2	FAU_SAR.1	
FAU_STG.1	FAU_GEN.1	
FDP_IFC.1	FDP_IFF.1	
FDP_IFF.1	FDP_IFC.1 FMT_MSA.3	
FIA_UAU.2	FIA_UID.2	FIA_UAU.2 has dependency on FIA_UID.1. Since FIA_UID.2 is hierarchical to that component the dependency is fulfilled.
FIA_UAU.6	None	
FIA_UID.2	None	
FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	
FMT_MSA.1	FDP_IFC.1 FMT_SMF.1 FMT_SMR.1	
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	
FMT_SMF.1	None	
FMT_SMR.1	FIA_UID.2	FMT_SMR.2 has dependency on FIA_UID.1. Since FIA_UID.2 is hierarchical to that component the dependency is fulfilled.
FPT_STM.1	None	
FTA_SSL.1	FIA_UAU.2	FTA_SSL.1 has dependency on FIA_UAU.1. Since FIA_UAU.2 is hierarchical to that component the dependency is fulfilled.
FTA_SSL.2	FIA_UAU.2	FTA_SSL.2 has dependency on FIA_UAU.1. Since FIA_UAU.2 is hierarchical to that component the dependency is fulfilled.

Table A.5: Dependencies of security functional requirements.

APPENDIX B

Security Target

Contents

B.1 ST Introduction

This section identifies the ST and the TOE to which it refers. Furthermore, the ST is summarized and the CC conformance claims are stated.

B.1.1 ST Identification

Title: Secure POS Interface for Payment Solutions Security Target

Publishing Date: 1st August 2005

ST Version Number: 1.0

Authors: Anders Hedegaard and Allan Pedersen

Version of CC used for development: CC version 2.2

Evaluation Assurance Level: EAL 3+

B.1.2 ST Overview

The TOE provides secure communication between a payment terminal and a POS application in a hosted Point of Sale system where the communication flows through an insecure path, e.g. the Internet. This scenario may arise when a merchant wants to integrate the handling of payment terminal transactions into a POS system application which is hosted at an other location by a POS system provider.

This ST describes the environment in which the TOE is to operate, the threats against it and the functionality required and provided to meet these threats.

B.1.3 CC Conformance

This Security Target is conformant to parts 2 and 3 of the Common Criteria v. 2.2 as follows.

- Part 2 conformant: The security functional requirements are based on those identified in part 2 of the Common Criteria.
- Part 3: The security assurance requirements are in the form of an EAL (assurance package) i.e. based upon assurance components in part 3 of the Common Criteria.

B.1.4 Abbreviations

The following abbreviations are used throughout the ST.

3DES Triple DES

AES Advanced Encryption Standard

CBC Cipher-Block-Chain

CC Common Criteria

DES Data Encryption Standard

EAL Evaluation Assurance Level

FIPS Federal Information Processing Standards

ICC Integrated Circuit Card

IT Information Technology

MSC Magnetic Stripe Card

OSP Organizational Security Policy

OTRS Open Terminal Requirements Specification

PA Payment Application

PAC Payment Application Client

PAN Primary Account Number

PIN Personal Identification Number

POS Point Of Sale

POSPP Point Of Sale CC Protection Profile

PP Protection Profile

PSAM Purchase Secure Application Module

RSA Rivest, Shamir and Adleman

SFP Security Function Policy

SFR Security Function Requirement

SOF Strength Of Function

ST Security Target

TAPA Terminal Architecture for PSAM Applications

TOE Target Of Evaluation

TSF TOE Security Function

TSP TOE Security Policy

B.2 TOE Description

The TOE described in this ST provides secure data flows between a payment terminal and a POS application, where the path of the data flows is insecure.

A common way of implementing a POS system is by hosting the POS application on a centralized server system and thereby providing access to the POS application and audit trail from the point of sale via a client-server based system. This means that the data flows to and from the input/output devices travel through the Internet, i.e. an insecure path, between the device and the POS application. The TOE is an interface between a payment terminal and a POS application in these types of POS systems.

B.2.1 TOE Boundaries and IT Environment

The device interface model from the POSPP is illustrated in figure B.1.

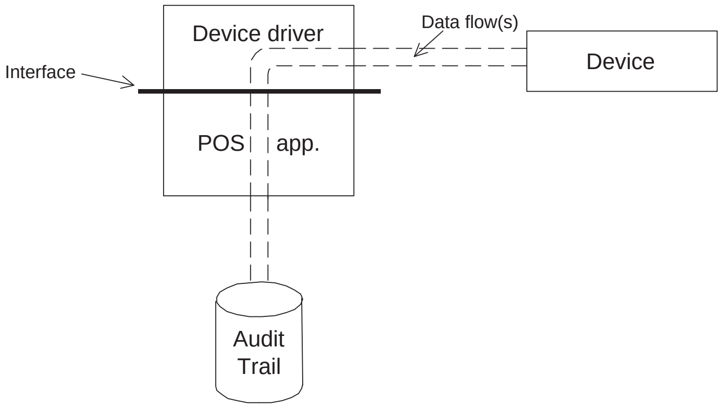

Figure B.1: TOE device interface.

Because the POS application and the payment terminal are separated it is necessary to refine this model to the one shown in figure B.2. The payment terminal is connected to the client-side of the POS system. Since the POS application is located at the server-side, the interface between the payment terminal and the POS application cannot be directly connected. The payment application located at the client-side of the POS system interfaces with the payment terminal. The POS application can initiate the data flows between itself and the payment application via a payment application client.

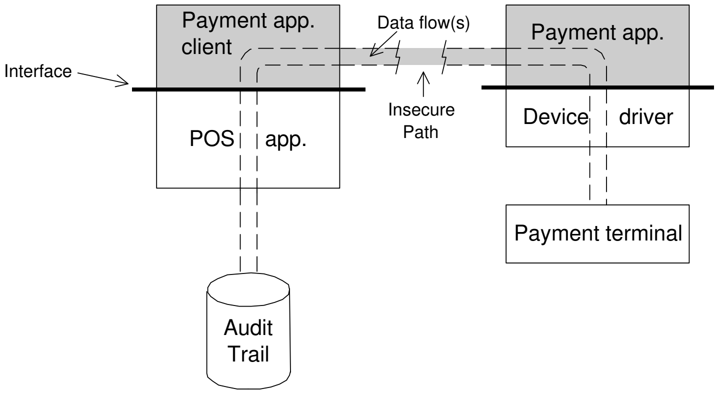

Figure B.2: TOE components and interfaces with POS application and device driver.

In addition the refined model defines the TOE boundaries. In figure B.2 the gray components indicate what is included in the TOE and the rest of the POS system is its IT environment. Hence, the TOE is bounded by the POS application's input/output device interfaces on one side and the device driver interface of the payment terminal on the other side.

The payment terminal may be any payment terminal complying with the Open Terminal Requirement Specification (OTRS) [PBS04] defined by PBS A/S [1] or any other terminal specification based on the Terminal Architecture for PSAM Applications (TAPA) [Eur01] equivalent to the OTRS.

The POS application may be any POSPP compliant POS system which is able to comply with the POS application interface provided by the TOE. As with the payment terminal, the POS application shall also comply with the requirements stated in OTRS. The TOE only secures the data flows between the two interfaces. How the POS application further processes the data is out of scope for this ST.

B.2.2 TOE Features

The TOE consists of two main components and is working as a client-server based system. The two components are the *Payment Application* (PA) and the *Payment Application Client* (PAC) as illustrated in figure B.2.

PA The Payment Application acts as the server part of the system. It is located at the device driver interface of the payment terminal, i.e. at the physically point of sale. The task of the PA is to monitor, audit and handle any event raising from the device driver of the payment terminal. When the PAC is connected the PA shall initiate the commands received from the PAC, and route the responses back to the PAC.

[1] `www.pbs.dk`

PAC The Payment Application Client acts as the client part of the system and implements the POS application interface. It is located at the POS application, which is physically separated from the point of sale when operating in a hosted environment as described earlier in section B.2. The task of the PAC is to establish a secure connection to the PA whenever the merchant wants to perform a transactional or administrative command on the payment terminal. When connected the PAC shall route the given command to the PA and likewise route any received events and responses back to the POS application via the POS application interface. The PAC shall terminate the secure connection when the command has been carried out and all responses are received.

B.2.3 Roles

From the POSPP four roles are mandatory to include. These are *Customer*, *Operator*, *Financial Manager*, and *Administrator*:

Customer Normally, the customer is assumed not to be authorized and interacts only with the TOE via the operator. Only in specialized operations of the POS system the customer may be asked to interact directly with the TOE, e.g. during a transaction via a payment terminal attached to the POS system, where the customer can be asked to swipe a credit card and enter a PIN.

Operator The operator is an authorized user of the TOE who is responsible for the actual registration and handling of the sold goods and received payments at the point of sale, e.g. a sales clerk. Furthermore, the operator is responsible for providing the produced evidence of the sale to the customer.

Financial Manager The financial manager is an authorized user of the TOE who is able to pull out information from the audit trail in relation to accounting.

Administrator The administrator is an authorized user of the TOE who is responsible for installation, configuration and maintenance of all functions of the POS system.

In addition, the definition of the role *Terminal Operator* from the OTRS ([PBS04] sec. 4.4.1) is needed.

Terminal Operator The terminal operator controls the PSAM (Purchase Secure Application Module) of the payment terminal and thereby the general functionality and security of the payment terminal. The terminal operator switches transactions from the terminal to the acquirer(s)[2] and resulting responses from acquirer(s) to the payment terminal.

[2] An acquirer is the authority responsible for obtaining transaction authorizations from the card issuers and for delivering settlement information to and from card issuers and merchants.

The role is needed to define the source and destination role of data flowing to and from the payment terminal to the POS application when analyzing the data flows of the TOE. In OTRS the customer and operator role is called *Card holder* and *Merchant* respectively.

B.2.4 Data Flows

The POSPP states that a data flow control policy shall be made for each data flow in the POS system. To do that a threat analysis must be carried out to determine the level of protection needed to secure the data flows. In the following the source and destination roles are indicated in parentheses. All data is flowing to and from the same input/output device, namely the payment terminal, and through the same media including a highly insecure path, the Internet. The following data flows are identified:

Transactional Commands (Operator → Terminal Operator) Operator initiated commands which cause a payment transaction to start, change state, or terminate.

Transactional Command Responses (Terminal Operator → Operator/Customer) Responses from transactional commands, which may be whether or not the commands was initiated successfully, and the result of the command. This includes actual transaction results and any transaction receipt for customer and operator.

Administrative Commands (Financial Manager → Terminal Operator) Commands initiated by the financial administrator to change the state of the terminal, flush data stores or request batch reports for totaling and financial postings.

Administrative Command Responses (Terminal Operator → Financial Manager) Responses from administrative commands, which may be whether or not the commands was initiated successfully, and the result of the command. This includes batch reports etc. if these are requested.

State Information Messages (Terminal Operator → Operator) These are the state information messages created by the PSAM and sent to the merchant display sub-handler of the merchant application handler in the TAPA model including action codes (see [PBS04] sec. 6.8.2).

Terminal Requests (Terminal Operator → Operator) Requests send from the PSAM for the operator to do manual verification actions. This is required during signature verification, during Integrated Circuit Card (ICC) to Magnetic Stripe Card (MSC) fall-back where the operator must verify that the card is inserted correctly, during manually card stop-list checks, and where the operator is requested to decide payment application selection or assist the customer in making the selection.

Terminal Request Responses (Operator → Terminal Operator) The response of the terminal requests from the operator including approval codes if necessary.

B.2.4.1 Sensitivity of Data

The sensitivity of monetary data will, by its nature, generally be high. The whole scheme of payment cards depends on the security by which the cards and their transactional data are handled. It is important that both the POS system owner and the customers find that it is a reliable and trustworthy method of payment.

The transactional and administrative commands and their responses contains sensitive personal data like the card Primary Account Number (PAN), amounts and dates related to every transaction processed. This type of data must be protected against disclosure to ensure the privacy of the customer and the financial data of the POS system owner. The terminal requests and their responses are sensitive because the security of the system relies on the integrity of the manual verifications they control - especially the signature verification. The state information messages does not as such contain sensitive data but the general functionality depends on the data and its integrity. A potential attacker may also find the terminal state as well as the terminal requests and their responses useful while trying to obtain the more sensitive data, hence these data flows shall also be protected against disclosure.

B.2.4.2 Threats Against the Data Flows

The most plausible threat agents of the data flows are unauthorized persons trying to intercept a data flow in order to collect sensitive data like card PANs or financial data. Card PANs can be abused in relation to direct credit card fraud if the attacker also collect other relevant data like PIN or expiry date of the card. The financial data can be abused in relation to industrial espionage. A more remote threat is an attacker intercepting the data flow trying to compromise the integrity of the data. This could be in changing the data in a way such that a different amount than expected is processed without the operator knowing it, or the attacker might intercept the terminal requests to fix the operator verifications.

B.2.4.3 Level of Protection

It is concluded that the appropriate level of protection of all the identified data flows is high, hence the most stringent and rigorous security countermeasures are required to protect a data flow between the payment application client and the payment application. This is due to the nature and sensitivity of the data in combination with the insecure path of the data flow. The data flow shall be implemented in a way that ensures the confidentiality and integrity of the data at a high level.

B.3 TOE Security Environment

B.3.1 Assumptions

This section describes the assumptions made for the TOE environment and the intended method of use of the TOE.

A.NO_EVIL It is assumed that the administrators of the TOE are competent of managing and maintaining the TOE and the security of the functions and data it contains. It is also assumed that these administrators do not have evil intentions of abusing their privileges.

A.THIRD_PARTY It is assumed that all third-party products used to implement the TOE environment (the general POS system, cryptographic service providers, etc) are trusted as well as correctly installed and configured.

B.3.2 Threats

This section describes the assets and threats of the TOE.

B.3.2.1 Assets

The primary asset to protect is the confidentiality and integrity of the data flows to and from the audit trail. If the data flows are disclosed, sensitive information may be collected and used with evil intentions by attackers. Furthermore, it can lead to incomplete financial accounting if the data is manipulated. Both things may conflict with legislation.

Secondary assets to protect are the security attributes of the TOE security functions (TSF) such as user names and passwords, cryptographic keys, etc.

B.3.2.2 Threat Agents

Threat agents are categorized as *authorized users* and *unauthorized users*. Note that the administrators are not considered threat agents due to the assumption **A.NO_EVIL**. In the following all threat agents are referred to as *attackers*.

Attackers are assumed to have various levels of expertise, motivation, and resources available. The expertise may come from specialized knowledge of the TOE. Motivation will normally arise from economic gain but personal revenge may also be a motivation.

B.3.2.3 Threats

The following threats are identified:

T.ACCESS An attacker may try to gain unauthorized access to the information protected by the TOE. This could be an unauthorized user impersonating an authorized user, or it may be an authorized user impersonating a, perhaps, more privileged user.

T.MODIFICATION An attacker may try to modify information protected by the TOE for which the attacker is not authorized.

T.PHYSICAL The audit trail may physically be lost due to fire, theft, force majeure, etc.

T.UNATTENDED_SESSION An attacker may gain unauthorized access to the TOE via a unattended session.

T.INCOMPETENCE A user may compromise the security of the TOE due to incompetent usage.

T.DATA_FLOW An attacker may compromise the confidentiality and integrity of an input or output data flow.

T.AUTHENTIC An attacker may try to impersonate the payment application or client, e.g. redirect the client connection to an unauthentic application.

T.CRYPTO_KEYS An attacker may compromise the security of the TOE by disclosing cryptographic keys in the TOE.

B.3.3 Organizational Security Policies

P.AUTHORIZED_USERS Only authorized users may access the TOE.

P.ACCOUNTABILITY The authorized users of the TOE shall be held accountable for their actions within the TOE.

P.TRAIN Any authorized user accessing security functions of the TOE shall receive continuous training in secure use of the TOE.

P.FIPS140 Any cryptographic function used by the TOE shall be FIPS 140 level 1 compliant.

B.4 Security Objectives

B.4.1 Security Objectives for the TOE

The following objectives states the security objectives for the TOE.

O.MANAGE The TOE shall provide functionality which enables authorized administrators to manage and support the security attributes of the TOE, and restrict these functions from unauthorized use.

O.AUDIT The TOE shall provide functionality to record security relevant events in sufficient detail to help administrators of the TOE to hold individual users accountable for any actions they perform that are relevant to the security of the TOE.

O.DATA_FLOW For attached input/output devices a data flow control policy based on a threat analysis shall be made for each identified data flow. This is done to accommodate the different demands to secure communication of the devices.

O.AUTHENTIC The payment application and payment application client shall perform mutual authentication before allowing any communication.

O.TRUSTED_CHANNEL The authentication process, session key distribution and communication of sensitive data shall be protected by a trusted channel between the payment application and payment application client.

B.4.2 Security Objectives for the Environment

For the general IT environment and POS IT environment the following objectives shall be met:

O.IA The TOE shall provide means for identifying and authenticating users before allowing access to the TOE and its resources.

O.SESSION A session shall only be active when an authorized user is interacting with the TOE interface. Therefore, the TOE shall provide functionality for the user to lock the current interactive session. It should also be possible for the TOE to automatically lock the session if the user is considered inactive. The user must re-authenticate to unlock the session. Furthermore, the user should re-authenticate before each sale and/or payment transaction.

O.BACK-UP The TOE shall provide functionality for the administrator to back up the data in the system in order to make it possible to restore, as a minimum, the audit trail in case of hacking, hardware failure, fire, theft, force majeure, etc.

O.FIPS140 The cryptographic service providers used to provide the cryptographic functions for the TOE shall be FIPS 140 validated to, at least, level 1.

For the general non-IT environment the following objectives shall be met:

OE.TRAIN The overall responsible for the TOE shall arrange training for all authorized users of the TOE including the administrators.

OE.ADMIN_VETTING The overall responsible for the TOE shall perform vetting of administrators to ensure that they are competent and non-hostile.

OE.PHYSICAL The TOE shall be physically protected in such a way that attackers cannot remove the TOE or parts of the TOE which are critical to the security of the TOE, or in other ways physically compromise the TOE and the data it contains, i.e. the audit trail, security attributes, etc.

B.5 IT Security Requirements

B.5.1 TOE Security Functional Requirements

This section describes the security functional requirements (SFR) components that shall be satisfied by the TOE. The components are taken from CC part 2 [CC204] and in table B.1 all identified SFRs are listed for a quick overview.

Class	**Family**	**Component**
FAU	FAU_GEN	FAU_GEN.1
FCS	FCS_COP	FCS_COP.1
FDP	FDP_IFC	FDP_IFC.1
	FDP_IFF	FDP_IFF.1
	FDP_ITT	FDP_ITT.1
		FDP_ITT.3
FMT	FMT_MOF	FMT_MOF.1
	FMT_MSA	FMT_MSA.1
		FMT_MSA.2
		FMT_MSA.3
	FMT_MTD	FMT_MTD.1
	FMT_SMF	FMT_SMF.1
FPT	FPT_ITT	FPT_ITT.1
FTP	FTP_ITC	FTP_ITC.1

Table B.1: SFR components.

In the following the TOE security functional requirements are listed in detail. The TSFs are listed in the same order as in the catalog. Text in *italic* indicates that an assignment, selection, or refinement have been performed.

B.5.1.1 FAU — Security Audit

FAU_GEN.1 Audit data generation

FAU_GEN.1.1(1) The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the *detailed* level of audit; and

c) *No other specifically defined auditable events.*

FAU_GEN.1.2(1) The TSF shall record within each audit record at least the following information:

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, *No other audit relevant information*

B.5.1.2 FCS — Cryptographic support

FCS_COP.1 Cryptographic operation

FCS_COP.1.1(1) The TSF shall perform *encryption of the payment application data flows and mutual authentication af payment application and payment application client* in accordance with *any of the following TLS cipher suites:*

a) TLS_RSA_WITH_3DES_EDE_CBC_SHA,

b) TLS_RSA_WITH_AES_128_CBC_SHA, or

c) TLS_RSA_WITH_AES_256_CBC_SHA.

and cryptographic key sizes *of 168 bit for 3DES, 128 or 256 bit for AES, and a minimum of 1024 bit for RSA* that meet the following:

a) FIPS 140 level 1 or equivalent.

Application note: *The TSF has been refined by removing the text "a specified cryptographic algorithm" for clarification. The actual cryptographic operations and its key management are assumed to be implemented in the general IT environment. The TOE shall ensure that the cryptographic service provider are used properly. The TLS ciphersuites are described in [DA99] and [Cho02].*

B.5.1.3 FDP — User Data Protection

FDP_IFC.1 Subset information flow control

FDP_IFC.1.1 The TSF shall enforce the *Payment Application Data Flow Control SFP* on *data flowing between the payment terminal and the POS application which causes information to flow into and out of the audit trail*

Application note: *The specific data flows are described in section B.2.4.*

FDP_IFF.1 Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the *Payment Application Data Flow Control SFP* based on the following types of subject and information security attributes:

a) Type of input/output device used in the data flow.

b) Role of user creating and receiving the data.

c) Type and sensitivity of the data.

d) Media in which the data flows.

e) Possible threat agents.

Application note: *The information security attributes listed are those used to conduct the threat analysis. The threat analysis is used to determine the level of protection the SFP needs to specify in order to achieve the desired security of the data flow.*

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:

a) A threat analysis of the input/output device data flow is carried out.

b) and the following countermeasures to achieve desired "high" level of protection for the data flow are implemented:

1) Secure authentication between the payment application and client ensuring correct authorization of the end points.

2) Encryption of the data flow using 3DES or AES ensuring the confidentiality and integrity of the data.

Application note: *The threat analysis is discussed in section B.2.4*

FDP_IFF.1.3 The TSF shall enforce *no addition information flow control rules*

Application note: *The TSF has been refined by removing the word "the" for clarification.*

FDP_IFF.1.4 The TSF shall provide *no addition SFP capabilities.*

Application note: *The TSF has been refined by removing the words "the following" for clarifications*

FDP_IFF.1.5 The TSF shall explicitly authorize an information flow based on the following rules: *None.*

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules: *None.*

FDP_ITT.1 Basic internal transfer protection

FDP_ITT.1.1 The TSF shall enforce the *Payment Application Data Flow Control SFP* to prevent the *disclosure and modification* of user data when it is transmitted between physically-separated parts of the TOE.

FDP_ITT.3 Integrity monitoring

FDP_ITT.3.1 The TSF shall enforce the *Payment Application Data Flow Control SFP* to monitor user data transmitted between physically-separated parts of the TOE for the following errors: *cryptographic integrity errors.*

FDP_ITT.3.2 Upon detection of a data integrity error, the TSF shall *try to resend the data up to a configurable number of times and alert the administrator.*

B.5.1.4 FMT — Security Management

FMT_MOF.1 Management of security functions behavior

FMT_MOF.1.1(1) The TSF shall restrict the ability to *determine the behaviour of, disable, enable and modify the behavior of* the functions*:*

a) The functions implementing the generation of security audit records, including which security events to record.

b) The functions controlling the behavior of the cryptografic functions, e.g. which cryptographic algorithm to use.

to *the administrators.*

FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the *Payment Application Data Flow Control SFP* to restrict the ability to *modify* the security attributes *referenced in the indicated policy* to *the administrators.*

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 initialization

FMT_MSA.3.1 The TSF shall enforce the *Payment Application Data Flow Control SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the *administrators* 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(1) The TSF shall restrict the ability to *change_default, query, modify, delete and clear* the

a) The audit data.

b) The cryptographic keys and attributes controlling the cryptographic functions.

to *the administrators*

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1(1) The TSF shall be capable of performing the following security management functions:

a) Functions to manage the audit behavior of the TOE.

b) Functions to manage the cryptographic functions.

B.5.1.5 FPT — Protection of the TSF

FPT_ITT.1 Basic internal TSF data transfer protection

FPT_ITT.1.1 The TSF shall protect TSF data from *disclosure and modification* when it is transmitted between separate parts of the TOE.

B.5.1.6 FTP — Trusted path/channels

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall provide a communication channel between *the payment application* and *the payment application client* that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

Application note: *The TSF has been refined by substituting "itself" with "the payment application" and "a remote trusted IT product" with "the payment application client".*

FTP_ITC.1.2 The TSF shall permit *the payment application client* to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for *authorization and communication.*

B.5.2 Security Requirements for the IT Environment

B.5.2.1 POS System IT Environment

As the TOE described in this ST is only a part or component of an entire POS system, the TOE will not comply with all security functional requirements stated in the POSPP. In order to claim conformance with the POSPP the ST shall accommodate any functional requirements from the POSPP which is not complied directly by the TOE by stating these as functional requirements for the IT environment, i.e. the rest of the POS system. The SFRs are summarized in table B.2.

Class	Family	Component
FAU	FAU_GEN	FAU_GEN.1
		FAU_GEN.2
	FAU_SAR	FAU_SAR.1
		FAU_SAR.2
	FAU_STG	FAU_STG.1
FIA	FIA_UAU	FIA_UAU.2
		FIA_UAU.6
	FIA_UID	FIA_UID.2
FMT	FMT_MOF	FMT_MOF.1
	FMT_MTD	FMT_MTD.1
	FMT_SMF	FMT_SMF.1
	FMT_SMR	FMT_SMR.1
FPT	FPT_STM	FMT_STM.1
FTA	FTA_SSL	FTA_SSL.1
		FTA_SSL.2

Table B.2: SFR components for the POS system IT environment.

In the following the SFRs of the IT environment to be encountered by the POS system are stated in detail. All open assigments from the POSPP are kept open, as they depend on how the general POS system is implemented. Consult the POSPP for guidance notes on these assigments.

B.5.2.2 FAU — Security Audit

FAU_GEN.1 Audit data generation

FAU_GEN.1.1(2) The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the [selection: choose one of: minimum, basic, detailed, not specified] level of audit; and

c) [assignment: other specifically defined auditable events].

FAU_GEN.1.2(2) The TSF shall record within each audit record at least the following information:

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [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_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide *the administrators* with the capability to read *any 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 from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to *prevent* unauthorized modifications to the audit records in the audit trail.

B.5.2.3 FIA — Identification and Authentication

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.6 Re-authenticating

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions:

a) The session has been locked or terminated.

b) A new sales and/or payment transaction is to be initiated.

c) [assignment: additional conditions under which re-authentication is required].

application note: *The conditions under which a session is locked or terminated is defined in FTA_SSL.*

FIA_UID.2 User identification before any action

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF mediated actions on behalf of that user.

B.5.2.4 FMT — Security Management

FMT_MOF.1 Management of security functions behavior

FMT_MOF.1.1(2) The TSF shall restrict the ability to [selection: determine the behaviour of, disable, enable, modify the behavior of] the functions*:*

a) The functions implementing the security auditing, including which security events to audit.

b) The functions implementing the input/output device data flow control policies for the attached input and output devices.

c) The functions implementing the method of identification and authentication of users.

d) The functions implementing timers and the clock synchronization.

e) The functions implementing the system backup routines.

f) The functions implementing the session locking methods.

g) [Assignment: additional manageable functions].

to *the administrators.*

FMT_MTD.1 Management of TSF data

FMT_MTD.1.1(2) The TSF shall restrict the ability to [selection: change_default, query, modify, delete, clear, [assignment: other operations]] the

a) The security audit trail.

b) The TOE system clock.

c) [Assignment: additional TSF data]

to *the administrators*

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1(2) The TSF shall be capable of performing the following security management functions:

a) Functions to assign and maintain lists of users and roles.

b) Functions to creating backups and recovering of, as a minimum, the audit trail.

c) Functions to set up and manage information flow controls for input and output devices.

d) Functions to manage the TOE system clock and timers.

e) Functions to manage and review the security audit trail.

f) Functions to manage session locking attributes.

g) [Assignment: other security management functions]

FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles:

a) Customer

b) Operator

c) Financial Manager

d) Administrator

e) Terminal Operator

f) [assignment: other identified roles].

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

Application note: *It may be argued that the Customer role normally do not need to have assigned users as they act as "anonymous" users identified and authorized by the Operator. See section B.2.3.*

B.5.2.5 FPT — Protection of the TSF

FPT_STM.1 Reliable time stamps

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

B.5.2.6 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:

a) clearing or overwriting display devices, making the current contents unreadable;

b) 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: *Re-authorization of user.*

FTA_SSL.2 User-initiated locking

FTA_SSL.2.1 The TSF shall allow user-initiated locking of the user´s own interactive session, by:

a) clearing or overwriting display devices, making the current contents unreadable;

b) disabling any activity of the user´s data access/display devices other than unlocking the session.

FTA_SSL.2.2 The TSF shall require the following events to occur prior to unlocking the session: *Re-authorization of user.*

Application note: *The time interval of user inactivity of the TSF-initiated session locking is to be defined with respect to the implementation of the user-initiated locking. If the locking is implemented in a way in such a way that the user cannot leave the TOE physically without locking a possible interactive session, e.g. with a smart-card attached to a key-chain, the time interval may be very long. It may even be infinite if it can be argued unnecessary due to stringent user-initiated locking functionality.*

B.5.2.7 General IT Environment

The Cryptographic Service Provider (CSP) used by the TOE for the cryptographic functions needed to protect the data flows are supposed to be provided by the general IT environment, the operating system (OS). Alternatively, the cryptographic functions may be provided by a third party CSP installed in the OS. The SFRs for the general IT environment are stated in the following.

Class	**Family**	**Component**
FCS	FCS_CKM	FCS_CKM.1
		FCS_CKM.2
		FCS_CKM.4
	FCS_COP	FCS_COP.1

Table B.3: SFR components for the general IT environment.

B.5.2.8 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 *SHS based random number generation as specified in FIPS 186 appendix 3 or a equivalent SHS based algorithm* and specified cryptographic key sizes *depending on cryptographic algorithm chosen for symmetric encryption of the payment application data flows* that meet the following:

a) FIPS 140 level 1 or equivalent.

Application note: *The list of approved cryptographic algorithms for symmetric encryption of the payment application data flows is specified in FCS_ COP.1.1(2).*

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 based key exchange* that meets the following:

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* cryptographic key destruction method that meets the following:

a) FIPS 140 level 1 or equivalent.

Application note: *The TSF has been refined by substituting the words "a specific" with "any" and, hence also removing the assignment of cryptographic key destruction method*

FCS_COP.1 Cryptographic operation

FCS_COP.1.1(2) The TSF shall perform *encryption of the payment application data flows and mutual authentication af payment application and payment application client* in accordance with *any of the following TLS cipher suites:*

a) TLS_RSA_WITH_3DES_EDE_CBC_SHA,

b) TLS_RSA_WITH_AES_128_CBC_SHA, or

c) TLS_RSA_WITH_AES_256_CBC_SHA.

and cryptographic key sizes *of 168 bit for 3DES, 128 or 256 bit for AES, and a minimum of 1024 bit for RSA* that meet the following:

a) FIPS 140 level 1 or equivalent.

Application note: *The TSF has been refined by removing the text "a specified cryptographic algorithm" for clarification. The TLS ciphersuites are described in [DA99] and [Cho02].*

B.5.3 TOE Security Assurance Requirements

This section describes the security assurance requirement. The assurance level is given by EAL3 from CC part 3 [CC304] with the addition of ADV_SPM.1, Informal TOE security policy model, to comply with the dependency of function requirement FMT_MSA.2, Secure security attributes.

Class	Component	
ACM	ACM_CAP.3	Authorization controls
	ACM_SCP.1	TOE CM coverage
ADO	ADO_DEL.1	Delivery procedures
	ADO_IGS.1	Installation, generation and start-up procedures
ADV	ADV_FSP.1	Informal functional specification
	ADV_HLD.2	Security enforcing high-level design
	ADV_RCR.1	Informal correspondence demonstration
	ADV_SPM.1	Informal TOE security policy model
AGD	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
ALC	ALC_DVS.1	Identification of security measures
ATE	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: high-level design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
AVA	AVA_MSU.1	Examination of guidance
	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.1	Developer vulnerability analysis

Table B.4: Security assurance components.

B.5.3.1 ACM — Configuration management

ACM_CAP.3 Authorization controls

Developer action elements

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

ACM_CAP.3.2D The developer shall use a CM system.

ACM_CAP.3.3D The developer shall provide CM documentation.

Content and presentation of evidence elements

ACM_CAP.3.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.3.2C The TOE shall be labeled with its reference.

ACM_CAP.3.3C The CM documentation shall include a configuration list and a CM plan.

ACM_CAP.3.4C The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.3.5C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.3.6C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.3.7C The CM system shall uniquely identify all configuration items.

ACM_CAP.3.8C The CM plan shall describe how the CM system is used.

ACM_CAP.3.9C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.3.10C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.3.11C The CM system shall provide measures such that only authorized changes are made to the configuration items.

Evaluator action elements

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

ACM_SCP.1 TOE CM coverage

Developer action elements

ACM_SCP.1.1D The developer shall provide a list of configuration items for the TOE.

Content and presentation of evidence elements

ACM_SCP.1.1C The list of configuration items shall include the following: implementation representation and the evaluation evidence required by the assurance components in the ST.

Evaluator action elements

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

B.5.3.2 ADO — Delivery and Operation

ADO_DEL.1 Delivery procedures

Developer action elements

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

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

Content and presentation of evidence elements

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

Evaluator action elements

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

ADO_IGS.1 Installation, Generation, and Start-up Procedures

Developer action elements

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

Content and presentation of evidence elements

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

Evaluator action elements

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

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

B.5.3.3 Development

ADV_FSP.1 Informal functional specification

Developer action elements

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

Content and presentation of evidence elements

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

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

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

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

Evaluator action elements

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

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

ADV_HLD.2 Security Enforcing High-level Design

Developer action elements

ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

Content and presentation of evidence elements

ADV_HLD.2.1C The presentation of the high-level design shall be informal.

ADV_HLD.2.2C The high-level design shall be internally consistent.

ADV_HLD.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_HLD.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

Evaluator action elements

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

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

ADV_RCR.1 Informal correspondence demonstration

Developer action elements

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

Content and presentation of evidence elements

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Evaluator action elements

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

ADV_SPM.1 Informal TOE security policy model

Developer action elements

ADV_SPM.1.1D The developer shall provide a TSP model.

ADV_SPM.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model.

Content and presentation of evidence elements

ADV_SPM.1.1C The TSP model shall be informal.

ADV_SPM.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

ADV_SPM.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

Evaluator action elements

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

B.5.3.4 AGD — Guidance Documents

AGD_ADM.1 Administrator guidance

Developer action elements

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

Content and presentation of evidence elements

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

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

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

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

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

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

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

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements

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

AGD_USR.1 User Guidance

Developer action elements

AGD_USR.1.1D The developer shall provide user guidance.

Content and presentation of evidence elements

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

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

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.

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

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements

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

B.5.3.5 ALC — Life cycle support

ALC_DVS.1 Identification of security measures

Developer action elements

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

Content and presentation of evidence elements

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

ALC_DVS.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

Evaluator action elements

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

ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

B.5.3.6 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 of evidence elements

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

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

Evaluator action elements

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

ATE_DPT.1 Testing: High-level Design

Developer action elements

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

Content and presentation of evidence elements

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

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 of evidence elements

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Evaluator action elements

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

ATE_IND.2 Independent Testing - Sample

Developer action elements

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

Content and presentation of evidence elements

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

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements

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

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

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

B.5.3.7 AVA — Vulnerability Assessment

AVA_MSU.1 Examination of guidance

Developer action elements

AVA_MSU.1.1D The developer shall provide guidance documentation.

Content and presentation of evidence elements

AVA_MSU.1.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.1.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.1.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.1.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

Evaluator action elements

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

AVA_MSU.1.2E The evaluator shall repeat all configuration and installation procedures to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.1.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_SOF.1 Strength of TOE Security Function Evaluation

Developer action elements

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

Content and presentation of evidence elements

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

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

Evaluator action elements

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

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

AVA_VLA.1 Developer vulnerability analysis

Developer action elements

AVA_VLA.1.1D The developer shall perform a vulnerability analysis.

AVA_VLA.1.2D The developer shall provide vulnerability analysis documentation.

Content and presentation of evidence elements

AVA_VLA.1.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2C The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.

AVA_VLA.1.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

Evaluator action elements

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

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

B.5.4 Strength of Function Claim

The strength of function claim will be that of SOF-medium as described in the CC part 1 [CC104].

B.6 TOE Summary Specification

B.6.1 TOE Security Functions

This section outlines the security functions supplied by the TOE to meet the security functional requirements for the TOE.

F.AUDIT The audit function assures that audit records are generated for each relevant security event and send to the audit trail. The POS application provides the functionality to store and review the generated records, as these functions are IT environment requirement.

F.CRYPTOGRAPHIC The cryptographic functions assures that a FIPS 140 level 1 compliant cryptographic service provider is used to implement a trusted channel between the payment application and client. Hence, the functions will ensure the confidentiality and integrity of the data flows.

F.MANAGEMENT The management functions assure that any configurable attribute or function are manageable by the administrator. The management functions also ensure that only an authorized administrator has access to these and that only secure attributes are accepted for the cryptographic functions.

B.6.2 Assurance Measures

This section specifies the assurance measures of the TOE which are claimed to satisfy the stated assurance requirements.

Component	How requirements will be met
ACM_CAP.3 Authorization controls	The developer will use a CVS to ensure that items placed in the CVS can only be modified in a controlled manner. Furthermore, the developer will provide CM documentation for the TOE.
ACM_SCP.1 TOE CM coverage	The developer will provide a list of of configuration items for the TOE.
ADO_DEL.1 Delivery procedures	The developer will use documented procedures, necessary to maintain security when distributing versions of the TOE, for delivery of the TOE.
ADO_IGS.1 Installation, generation and start-up procedures	The developer will provide documentation describing steps needed for secure installation.

Continued on next page

Component	**How requirements will be met**
ADV_FSP.1 Informal functional specification	The developer will provide a functional specification describing the TSF and its external interfaces in an informal style.
ADV_HLD.2 Security enforcing high-level design	The developer will provide an informal high-level design of the TSF.
ADV_RCR.1 Informal correspondence demonstration	The developer will provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided, i.e. TOE summary specification, high- and low-level design, and implementation representation.
ADV_SPM.1 Informal TOE security policy model	The developer will demonstrate correspondence between the functional specification and the informal TSP model.
AGD_ADM.1 Administrator guidance	The developer will provide administrator guidance describing how to administer the TOE in a secure manner.
AGD_USR.1 User guidance	The developer will provide user guidance.
ALC_DVS.1 Identification of security measures	The developer will produce development security documentation describing all security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ATE_COV.2 Analysis of coverage	The developer will provide an analysis of the test coverage.
ATE_DPT.1 Testing: high-level design	The developer will provide the analysis of the depth of testing.
ATE_FUN.1 Functional testing	The developer will test the TSF and document the results. Test documentation consisting of test plans, test procedure descriptions, expected test results, and actual test results will be provided by the developer.

Continued on next page

Component	**How requirements will be met**
ATE_IND.2 Independent testing - sample	The developer will provide the TOE for testing and the means for the evaluator to recreate the conducted tests, and conduct additional tests.
AVA_MSU.1 Examination of guidance	The developer will provide guidance documentation identifying all possible modes of operation of the TOE, their consequences, and implications for maintaining secure operation.
AVA_SOF.1 Strength of TOE security function evaluation	The developer will perform a SOF analysis to prove that each mechanism identified have the claimed SOF.
AVA_VLA.1 Developer vulnerability analysis	The developer will perform and document a vulnerability analysis showing that a vulnerability cannot be exploited in the intended environment.

Table B.5: Assurance measures.

B.7 PP Claims

B.7.1 PP Reference

The TOE is conformant with the Point of Sale System CC Protection Profile (POSPP) version 1.0.

B.7.2 PP Tailoring and Additions

Table B.6 shows modifications to the assumptions, threats, and OSPs relative to the POSPP. The following modifications may be performed: *None* means no modification relative to the POSPP, *Added* means new component is added relative to the POSPP, and *Enhanced* means enhancement of the component relative to the POSPP.

Name	**Modification**
A.NO_EVIL	None
A.THIRD_PARTY	Added
T.ACCESS	None
T.MODIFICATION	None
T.PHYSICAL	None
T.UNATTENDED_SESSION	None
T.INCOMPETENCE	None
T.DATA_FLOW	Enhanced
T.AUTHENTIC	Added
T.CRYPTO_KEY	Added
P.AUTHORIZED_USERS	None
P.ACCOUNTABILITY	None
P.TRAIN	None
P.FIPS	Added

Table B.6: Modifications to assumptions, threats, and OSPs relative to the POSPP.

The modifications performed to the components are conducted in order to clarify the new specific issues of the TOE.

Table B.7 shows modifications to the objectives relative to the POSPP. The modification *Moved* means the component has been changed from being a TOE security objective to being a security objective for the IT environment. Objectives are added in order to counter the new and enhanced assumptions, threats, and OSPs.

Table B.8 shows modifications to the SFRs relative to the POSPP. *Assignment* means one or more assignments or selections have been performed on the SFR. *Iterated* means the SFR components have been iterated in order to specify that the SFR shall be countered partly by the TOE and partly by the IT environment.

Name	Modification
O.MANAGE	None
O.AUDIT	None
O.DATA_FLOW	None
O.AUTHENTIC	Added
O.TRUSTED_CHANNEL	Added
O.IA	Moved
O.SESSION	Moved
O.BACK-UP	Moved
O.FIPS140	Added
OE.TRAIN	None
OE.ADMIN_VETTING	None
OE.PHYSICAL	None

Table B.7: Modifications relative to POSPP.

No modifications have been conducted to the assurance requirements except for the addition of the assurance requirement ADV_SPM.1. The addition has been performed in order to comply with the dependency of the functional requirement FMT_MSA.2.

Class	Modification
FAU_GEN.1	Assigned Iterated
FAU_GEN.2	Moved
FAU_SAR.1	Moved
FAU_SAR.2	Moved
FAU_STG.1	Moved
FCS_CKM.1	New
FCS_CKM.2	New
FCS_CKM.4	New
FCS_COP.1	New
FDP_IFC.1	Assignment
FDP_IFF.1	Assignment
FDP_ITT.1	New
FDP_ITT.3	New
FIA_UAU.2	Moved
FIA_UAU.6	Moved
FIA_UID.2	Moved
FMT_MOF.1	Assignment Iterated
FMT_MSA.1	Assignment
FMT_MSA.2	New
FMT_MSA.3	Assignment
FMT_MTD.1	Assignment Iterated
FMT_SMF.1	Assignment Iterated
FMT_SMR.1	Moved Assignment
FPT_ITT.1	New
FPT_STM.1	Moved
FTA_SSL.1	Moved
FTA_SSL.2	Moved
FTP_ITC.1	New

Table B.8: Modifications to SFRs relative to the POSPP.

B.8 Rationale

B.8.1 Security Objective Rationale

Table B.9 illustrates the relations between the security objectives and the assumptions, threats and OSPs they counter.

	O.MANAGE	O.AUDIT	O.DATA_FLOW	O.AUTHENTIC	O.TRUSTED_CHANNEL	O.IA	O.SESSION	O.BACK-UP	O.FIPS140	OE.TRAIN	OE.ADMIN_VETTING	OE.PHYSICAL
A.NO_EVIL										x	x	
A.THIRD_PARTY									x	x	x	
T.ACCESS	x	x		x		x						
T.MODIFICATION	x	x				x		x				
T.PHYSICAL								x				x
T.UNATTENDED_SESSION							x			x		
T.INCOMPETENCE										x		
T.DATA_FLOW	x		x	x	x				x			
T.AUTHENTIC				x	x				x			
T.CRYPTO_KEYS				x	x				x			
P.AUTHORIZED_USERS	x					x						
P.ACCOUNTABILITY	x	x				x	x					
P.TRAIN										x		
P.FIPS140									x			

Table B.9: Relations illustrating the security objective rationale.

B.8.1.1 Security Objectives Suitable to Uphold Assumptions

The following rationale demonstrates how the objectives cover the assumptions:

A.NO_EVIL OE.TRAIN upholds this assumption because it assures that the administrators stay competent via continuous training.

OE.ADMIN_VETTING upholds this assumption because it ensures that all administrators are vetted to ensure that they stay competent and non-hostile.

A.THIRD_PARTY OE.TRAIN upholds this assumption because it assures that the administrators stay competent via continuous training thereby assuring that they are able to install third party products and still upholding security.

OE.ADMIN_VETTING upholds this assumption because it ensures that all administrators are vetted to ensure that they stay competent and non-hostile.

O.FIPS140 upholds this assumption as it ensures that if a third party product includes cryptographic functions they are, at least, FIPS140 level 1 validated.

B.8.1.2 Security Objectives Suitable to Counter the Threats

The following rationale demonstrates how the objectives counter the threats:

T.ACCESS This threat is mainly countered by O.IA which provides the means to identify and authenticate users before they are granted access to the TOE. In this way services of the TOE are only available if a user is identified which reduces the threat of unauthorized access to the TOE.

O.AUTHENTIC ensures that the PAC is connected to the authentic PA. This assures that a user is not connected to an unauthentic server when trying to send data.

O.MANAGE restricts the use of TOE security functions from unauthorized use. This helps to reduce the threat of unauthorized access to the TOE.

O.AUDIT provide the means to record security relevant actions in the TOE. In this way administrators are able to keep track of user actions in the TOE and they can take necessary actions if suspicious activities are recorded.

T.MODIFICATION This threat is mainly countered by O.IA which provides the means to identify and authenticate users before they are granted access to the resources of the TOE, and O.MANAGE which makes the administrators capable of restricting unauthorized use of security related functions.

O.AUDIT provides means to record security relevant actions, e.g new entries in the audit trail or modification of security attributes, in the TOE. In this way unauthorized modification of data in the TOE is tracked and it is possible to restore previous conditions. If more critical modifications have maliciously been performed a backup recovery operation may be required. This is addressed by OE.BACK-UP but should only be used when no other options are available.

T.PHYSICAL This threat is mainly countered by O.BACK-UP which states that administrators shall back up, as a minimum, the audit trail in order to make it possible to restore the data in case of physical loss in case of fire, theft, force majeure, etc. This objective effectively decreases the threat of physical loss, especially if a good back up plan is made.

OE.PHYSICAL states that the TOE shall be physically protected in a way that attackers cannot remove the TOE or in any other way physically compromise it or the data it contains. By securing the TOE physically, e.g. bolting it to the ground, the threat of physical loss is reduced.

T.UNATTENDED_SESSION This threat is mainly countered by O.SESSION which ensures that a session cannot be left unattended. It makes it possible for an authorized user to lock the the current session. If the user leaves the session without locking it the TOE automatically locks the session.

OE.TRAIN ensures that all authorized users of the TOE receives continuously training in use of the TOE. Education reduces the risk of users leaving an open session and thereby leaving the TOE open for attackers.

T.INCOMPETENCE This threat is countered by OE.TRAIN which ensures that all authorized users are continuously trained and educated in secure use of the TOE. This will effectively reduce the threat of users using the TOE in an incompetent way which compromises the security of the TOE.

T.DATA_FLOW This threat is mainly countered by O.DATA_FLOW which ensures that a threat analysis is carried out for each data flow and suitable security measures are implemented.

O.AUTHENTIC and O.TRUSTED_CHANNEL ensure that both PA and PAC are mutually authenticated before a data flow is initialized. Furthermore, they assure that this data is protected by a trusted channel.

O.FIPS140 ensures that cryptographic functions are, at least, FIPS140 level 1 validated.

O.MANAGE ensures that administrators are able to manage TOE security functions in a secure way. In this way the data integrity is preserved and the risk of attacks on the data flows is minimized.

T.AUTHENTIC This threat is mainly countered by O.AUTHENTIC and O.TRUSTED_CHANNEL which ensure that the PA and the PAC are mutually authenticated through a trusted channel before allowing any communication.

O.FIPS140 ensures that cryptographic functions are, at least, FIPS140 level 1 validated.

T.CRYPTO_KEYS This threat is mainly countered by O.AUTHENTIC and O.TRUSTED_CHANNEL which ensure that the PA and the PAC are communicating through a trusted channel and thereby minimizing the risk of cryptographic keys disclosure.

O.FIPS140 ensures that cryptographic functions are, at least, FIPS140 level

1 validated.

O.MANAGE makes the administrators capable of managing cryptographic keys and functions.

B.8.1.3 Security Objectives Suitable to Meet OSPs

The following rationale demonstrates how the objectives achieve the OSPs:

P.AUTHORIZED_USERS O.IA ensures that the TOE supports authentication and identification of users before they gain access to the TOE. In this manner only authorized users are able to access the TOE.

O.MANAGE ensures that security functions are managed in a way in such a way that only authorized users have access to these.

P.ACCOUNTABILITY O.IA, O.SESSION, and O.AUDIT ensure that users are held responsible for their actions in the TOE. This is because users have to (re-)authenticate and identify themselves before the TSFs allow any action in the TOE. Furthermore, all security relevant actions are recorded which enables administrators to monitor any unusual traffic.

O.MANAGE ensures that security functions are managed in a way which assures that all security relevant actions are recorded.

P.TRAIN OE.TRAIN ensures that authorized users of the TOE receive continuous training in secure use of the TOE.

P.FIPS140 O.FIPS140 ensures that all cryptographic functions are, at least, FIPS140 level 1 validated.

B.8.2 Security Requirements Rationale

Table B.10 provides the correlation between the security objectives to be met by the TOE.

Security Objective	Security Functional Requirement
O.MANAGE	FMT_MOF.1 FMT_MSA.1 FMT_MSA.2 FMT_MSA.3 FMT_MTD.1

Continued on next page

Security Objective	Security Functional Requirement
	FMT_SMF.1 FMT_SMR.1
O.AUDIT	FAU_GEN.1 FAU_GEN.2 FAU_SAR.1 FAU_SAR.2 FAU_STG.1 FMT_UID.2 FMT_SMF.1 FMT_SMR.1 FPT_STM.1
O.DATA_FLOW	FCS_CKM.1 FCS_CKM.2 FCS_CKM.4 FCS_COP.1 FDP_IFC.1 FDP_IFF.1 FDP_ITT.1 FDP_ITT.3 FPT_ITT.1 FTP_ITC.1
O.AUTHENTIC	FDP_ITT.1 FDP_ITT.3 FPT_ITT.1 FTP_ITC.1
O.TRUSTED_CHANNEL	FCS_CKM.1 FCS_CKM.2 FCS_CKM.4 FCS_COP.1 FTP_ITC.1
O.IA	FIA_UAU.2 FIA_UAU.6 FIA_UID.2 FMT_SMF.1 FMT_SMR.1
O.SESSION	FIA_UAU.6 FTA_SSL.1 FTA_SSL.2

Continued on next page

Security Objective	Security Functional Requirement
O.BACK-UP	FMT_SMF.1
O.FIPS140	FCS_COP.1 FCS_CKM.1 FCS_CKM.2 FCS_CKM.4

Table B.10: Security requirements rationale.

O.MANAGE The components FMT_MSA.1, FMT_MOF.1, FMT_MTD.1, and FMT_SMR.1 ensure that only authorized users (roles) are able to manage the security attributes. FMT_SMF.1 provides the security functions to manage the attributes.

FMT_MSA.2 ensures that only secure values are accepted for security attributes.

FMT_MSA.3 ensures that the TSF provides default values for relevant security attributes.

O.AUDIT The component FAU_GEN.1 ensures that auditable events are identified for which audit records should be generated and which information the records contain, e.g. user log-out. FAU_GEN.2 associates users with each record.

The components FAU_SAR.1 and FAU_SAR.2 ensures that only users that have been granted explicit read-access are able to read audit records.

FAU_STG.1 ensures that the TSF shall protect the stored audit records from unauthorized deletion. Furthermore the TSF shall prevent unauthorized modifications to the audit records in the audit trail.

The components FMT_UID.2 and FMT_SMR.1 ensures that users (roles) are identified and authenticated before they can interact with the TOE. This makes the previously mentioned user association possible.

FMT_SMF.1 provides the security functions to manage the attributes.

The component FPT_STM.1 ensures that TSFs provide reliable time stamps for its own use. This is necessary to make sure that audit records in the audit trail are reliable.

O.DATA_FLOW The component FDP_IFC.1 ensures that an information flow control SFP is made the identified data flows. FDP_IFF.1 ensures that the data flows are secured at a high level of protection as determined by the threat analysis.

FTP_ITC.1, FPT_ITT.1, and FDP_ITT.1 ensure that TSF and user data are protected against disclosure and modification when communicating between PA and PAC. FTP_ITT.3 ensures monitoring of integrity and actions in case of detection of errors.

FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, and FCS_COP.1 ensures that that cryptographic functions and PA/PAC authentication are, at least, FIPS140 level 1 validated.

O.AUTHENTIC FTP_ITC.1 and FPT_ITT.1 ensure that the PA and the PAC are mutually authenticated and that TSF data is not disclosed or modified during the authentication.

O.TRUSTED_CHANNEL FTC_ITC.1 provides a trusted communication channel between the PAC and the PA. FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, and FCS_COP.1 ensure that cryptographic operations are, at least, FIPS140 validated.

O.IA The components FIA_UAU.2, FIA_UID.2, and FMT_SMR.1 ensure that users (roles) have to identify and authenticate themselves before any action in the TOE is allowed by the TSF. On special occasions it may be necessary to re-authenticate a user, e.g. on session time-outs. Re-authentication is ensured by FIA_UAU.6.

FMT_SMF.1 provides the security functions to manage the attributes.

O.SESSION The components FIA_UAU.6, FTA_SSL.1, and FTA_SSL.2 ensures that both TSF- and user-initiated session locking are possible and it requires re-authentication to unlock the session.

O.BACK-UP FMT_SMF.1 provides the security functions to manage the back-up attributes.

O.FIPS140 FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, and FCS_COP.1 ensure that cryptographic operations are, at least, FIPS140 validated.

B.8.2.1 Dependencies of Security Requirements

Table B.11 gives all dependencies met by the SFRs and thereby proofs that the SFRs are mutually supportive and internally consistent.

SFR	**Dependency**	**Note**
FAU_GEN.1	FPT_STM.1	
FAU_GEN.2	FAU_GEN.1 FIA_UID.2	FAU_GEN.2 has dependency on FIA_UID.1. Since FIA_UID.2 is hierarchical to that component the dependency is fulfilled.
FAU_SAR.1	FAU_GEN.1	
FAU_SAR.2	FAU_SAR.1	
FAU_STG.1	FAU_GEN.1	
FCS_CKM.1	FCS_COP.1 FCS_CKM.4 FMT_MSA.2	
FCS_CKM.2	FCS_CKM.1 FCS_CKM.4 FMT_MSA.2	
FCS_CKM.4	FCS_CKM.1 FMT_MSA.2	
FCS_COP.1	FCS_CKM.1 FCS_CKM.4 FMT_MSA.2	
FDP_IFC.1	FDP_IFF.1	
FDP_IFF.1	FDP_IFC.1 FMT_MSA.3	
FDP_ITT.1	FDP_IFC.1	
FDP_ITT.3	FDP_IFC.1 FDP_ITT.1	
FIA_UAU.2	FIA_UID.2	FIA_UAU.2 has dependency on FIA_UID.1. Since FIA_UID.2 is hierarchical to that component the dependency is fulfilled.
FIA_UAU.6	None	
FIA_UID.2	None	
FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	
FMT_MSA.1	FDP_IFC.1 FMT_SMF.1 FMT_SMR.1	
FMT_MSA.2	ADV_SPM.1	

Continued on next page

SFR	Dependency	Note
	FDP_IFC.1 FMT_MSA.1 FMT_SMR.1	
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	
FMT_SMF.1	None	
FMT_SMR.1	FIA_UID.2	FMT_SMR.2 has dependency on FIA_UID.1. Since FIA_UID.2 is hierarchical to that component the dependency is fulfilled.
FPT_ITT.1	None	
FPT_STM.1	None	
FTA_SSL.1	FIA_UAU.2	FTA_SSL.1 has dependency on FIA_UAU.1. Since FIA_UAU.2 is hierarchical to that component the dependency is fulfilled.
FTA_SSL.2	FIA_UAU.2	FTA_SSL.2 has dependency on FIA_UAU.1. Since FIA_UAU.2 is hierarchical to that component the dependency is fulfilled.
FTP_ITC.1	None	

Table B.11: Dependencies of security functional requirements.

B.8.3 TOE Summary Specification Rationale

B.8.3.1 IT Security Functions

Table B.12 provides the correlation between the IT security functions and the security functional requirements. It that the IT security functions stated satisfy all the security functional requirements for the TOE.

IT Security Function	SFRs
F.AUDIT	FAU_GEN.1 and FAU_GEN.2
F.CRYPTOGRAPHIC	FCS_COP.1, FDP_IFC.1, FDP_IFF.1, FDP_ITT.1, FDP_ITT.3, FPT_ITT.1, and FTA_ITC.1
F.MANAGEMENT	FMT_MOF.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1, and FMT_SMF.1

Table B.12: Mapping of IT security functions to SFRs.

B.8.3.2 Strength of Function Claims

The Strength of Function claim made for this ST is SOF-medium. This is considered adequate because the TOE has a medium attack potential. Furthermore, this SOF level provides adequate protection against straightforward or intentional breach which is believed to be the most plausible attacks against the TOE.

B.8.3.3 Assurance Measures

The assurance level of this ST is EAL3 augmented with ADV_SPM.1. Table B.5 shows that assurance measures are compliant with the assurance requirements.

B.8.4 PP Claims Rationale

This ST claims conformance to the POSPP version 1.0. Section B.7.2 states any modification and additions made to security objectives and security requirements relative to the POSPP. Additions and enhancements are considered not to conflict with the PP conformance claims as they do not reduce the PP security requirements.

As the TOE is only a component of a POS system it cannot comply with all objectives and requirements stated in the POSPP. But by moving the objectives and requirements, not directly countered by the TOE, to the IT environment a fully POSPP compliant POS system is achieved. The moving is done in accordance with the application notes in section A.6 in the POSPP.

Some of the objectives and requirements are met partly by the TOE and partly by the IT environment. In these cases an iteration of the requirement has been performed in order to state the requirement for both the TOE and the IT environment. As this is not a reduction of the requirements it will not conflict with the POSPP requirements.

All assignments and selections of the TOE security functional requirements have been conducted. Any open assignment and selection from the POSPP countered by the IT environment are left open to be specified by the implementors of the remaining parts of the POS system.