Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Sichere Software

Common Criteria Protection Profile

Discretionary
Information Flow Control (SU)

Discretionary Information Flow Control

Certification ID	BSI-PP-0007
Identification	01345-DIC-SU
Version no.	2.01
Date	September 4, 2002
Authors	Dr. Steffen Lange Dr. Andreas Nonnengart Christian Stüble Roland Vogt

Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI GmbH),
Transferzentrum Sichere Software, Stuhlsatzenhausweg 3, D-66123 Saarbrücken

Table of Contents

List of Figures

List of Tables

1 PP Introduction

1.1 PP Identification

Title: Discretionary Information Flow Control (SU)

Version: 2.01

Registration: Bundesamt für Sicherheit in der Informationstechnik (BSI)
[German Information Security Agency]

Certification ID: BSI-PP-0007

This protection profile is hierarchically below the protection profile "Discretionary Information Flow Control", Certification ID: BDI-PP-0008 [BISS-MU].

This protection profile has been drawn up on the basis of:

- Common Criteria for Information Technology Security Evaluation, Part 1-3, Version 2.1, August 1999
- Common Methodology for Information Technology Security Evaluation
  - Part 1, Version 0.6, January 11, 1997
  - Part 2, Version 1.0, August 1999
- CCIMB Final Interpretations, Issue February 15, 2002
- ISO-Guide for the Production of Protection Profiles and Security Targets, Version 0.9, January 4, 2000
- Anwendungshinweise und Interpretationen zum Schema, AIS32, Übernahme international abgestimmter CC-Interpretationen ins deutsche Zertifizierungsschema, Version 1, 02.07.2001, Bundesamt für Sicherheit in der Informationstechnik

1.2 PP Overview

The task of the TOE is to protect the information flows of an IT system transparently for the users. That is why the TOE controls the admission of an information flow according to definable information flow rules. The security service supports in particular those IT users with little technical competence in asserting security of information as far as the aspects confidentiality, integrity and/or authenticity are concerned. The TOE security features can be considered as a useful addition to well-established security concepts such as access control, transmission protection, firewalls or Virtual Private Networks. The TOE can be used in areas like:

- E-commerce (data warehouses etc.),
- E-government (tendering procedures, application procedures etc.),
- Health-care system (electronic patient records etc.) and
- Tele and media services (teleworking etc.).

To every information flow a combination of security mechanisms can be allocated that corresponds to this information flow's security requirement. For the controlled information these mechanisms guarantee selectively the protection of the

- integrity by electronic signature,
- confidentiality by encryption,
- authenticity by electronic certificates

In this context the maintenance of confidentiality serves to avoid undesired knowledge of locally stored user data (e.g. when data carriers are stolen or the processing is inappropriate) and of user data during the transmission of messages. Integrity and authenticity are particularly important when it comes to commercial transactions (e.g. electronic orders and electronic payments).

Another protection mechanism is the restriction of information processing to certain subjects (e.g. applications). This supports to realise in a technical sense the appropriateness of information processing in accordance with privacy protection regulations.

The TOE operates almost completely transparent for the concerned subjects (e.g. applications) and for IT system users. The applications employed in the IT system only has to be adapted, if at all, in such a way that the TOE obtains the information about the corresponding information flows that is needed in order to maintain control keeping. Flexible configuration options enable an individual and consistent adjustment of the TOE to the protection requirements of the IT system operator.

The protection profile abstracts from technical details such that the TOE can be realised for various IT-environments such as e.g.

- operating systems
- database systems or
- e-mail clients and e-mail servers.

The security feature described here assumes that there is no need to distinguish users. In cases where the IT environment supports the distinction of users and where it is advisable to exploit this ability, the use of a TOE is recommended that is conformant with the multi user variant [DIC-MU] of the protection profile.

The TOE can be integrated in the respective IT environment in many different ways. For instance, a service process of an operating system such as an e-mail server can be considered as a single subject being controlled by a TOE which is integrated in the operating system. On the other hand, this service process can also be considered as an independent IT environment for a TOE which controls clients communicating with the server.

1.3 PP Organisation

The main sections of the PP are the TOE description, TOE security environment, security objectives, IT security requirements, and rationale.

The TOE description provides general information about the TOE, serves as an aid to understanding its security requirements, and provides context for the PP's evaluation. Both the product type and the general TOE functionality are described. In the sections TOE Boundary and TOE Operational Environment the main components of the TOE are identified and its embedding in the IT environment is described. In order to help the reader to understand the security concepts of the TOE, basic notions are introduced in section TOE Security Policy. This is followed by a detailed description of the security principles and security characteristics of the "discretionary information flow control" security functional policy (SFP).

The TOE security environment describes security aspects of the environment in which the TOE is to be used and the manner in which it is to be employed. The TOE security environment includes descriptions of

a) assumptions regarding the TOE's intended usage and environment of use
b) threats relevant to secure TOE operation, and
c) organisational security policies with which the TOE must comply.

The security objectives reflect the stated intent of the PP (independently from any product). They pertain to how the TOE will counter identified threats and how it will cover identified organisational security policies and assumptions. Each security objective is categorised as being for the TOE or for the environment.

The IT security requirements section provides detailed security requirements, in separate subsections, for the TOE and its environment. The TOE security requirements are subdivided as follows:

a) TOE security functional requirements including strength of function requirements for TOE security functions realised by a probabilistic or permutational mechanism, and
b) TOE security assurance requirements.

The Rationale presents evidence that the PP is a complete and cohesive set of IT security requirements and that a conformant TOE would effectively address the security needs. The Rationale is factored into two main parts. First, a Security Objectives Rationale demonstrates that the stated security objectives are traceable to all of the aspects identified in the TOE security environment and are suitable to cover them. Then, a Security Requirements Rationale demonstrates that the security requirements (TOE and environment) are traceable to the security objectives and are suitable to meet them.

2 TOE Description

Products which are in conformant with this protection profile generally consist of one or more component that extend the already existing input/output functions of the IT system by a user definable information flow control. The requirements for the TOE are kept on a very general level so that the creation of conformant products is possible for various IT environments. The TOE can be used for different purposes. For instance, it allows for the protection of local data and their processing as well as for the protection of data that have to be transferred via open networks.

Information flows occurring within the IT system such as storing or loading of data files as well as information flows leaving the IT system such as sending and receiving of e-mails are controlled. In order to identify information flows the processing functional unit (e.g. an application) and the data depositories (e.g. directories, computer addresses, and e-mail addresses) are used. Plausibility and consistency checks prevent that information flow rules are inappropriate or contradictory. Both also help to ease the administration of the TOE.

The TOE is always invisible for the user. It is only perceived in case of error messages, if signing/encrypting processes are applied, and if information flows are to be authorised explicitly.

2.1 Product Type

The TOE is either a component of an IT system which is close to the operating system or it is a proper component of an IT system's operating system. For realising the TSF a modular architecture is advisable which enables the integration of the TOE with different application services such as database systems and e-mail services. The TOE can be realised either as a pure software solution or as a combined solution consisting of software and hardware components. Particularly for the management and the application of cryptographic keys the TOE may use suitable (hardware) modules in the IT environment.

2.2 IT Features

The TOE ensures that information flows that occur within the IT system or that leave the IT system (e.g. via open nets such as LAN, WAN, internet, e-mail) obey the underlying TOE security policy and the specified information flow rules. The information flow rules can be derived from given legal, technical and organisational regulations (e.g. encryption of stored data, encryption and signing of transferred data). In information flow rules it can be specified under which circumstances and in which manner the TOE has to proceed the data. With the help of a reference monitor that supervise all information flows, the TOE decides about the permission/rejection of information flows and the approved processing of the data.

2.3 TOE Boundary

The TOE comprises a constantly active functional processing component which consists of a reference monitor, control and processing functions as well as a list of information flow rules. The functional processing component supervises and processes the information flows. In addition, there are functions for configuration, administration and the evaluation of protocols.

Application note 1. The ST author is in charge of describing the components of the TOE more closely, for example when an independent program is to be used for the administration.

This protection profile is presented in a way such that different architectures are possible for the realisation of products:

- Component TOE – The functions for information flow control are separated from the functions for management and application of cryptographic keys. This enables in particular the usage of prefabricated crypto modules.
- Composite TOE – The functions for information flow control as well as for the management and application of cryptographic keys are an integral part of the product.

The IT security requirements are specified in such a way that the protection profile is directly suitable for the realisation of a component TOE, which relies on external services for cryptographic support. For the conformity of a composite TOE, that has this functionality as an integral part, the required functional components have to be moved to the section TOE Security Requirements. In particular this is true for the component FCS_COP.1 which, in this case, has to be considered as a requirement for the TOE instead of a requirement for the IT environment.

An admissible structure of the TOE and its integration in the IT environment is shown in Figure 1 in form of a block diagram. The coloured area marks the TOE. The basic module provides the basic functionality (reference monitor). The extension modules connect different application services (e.g. database systems and e-mail-services) as well as the functions for configuration, administration and evaluation of protocols.

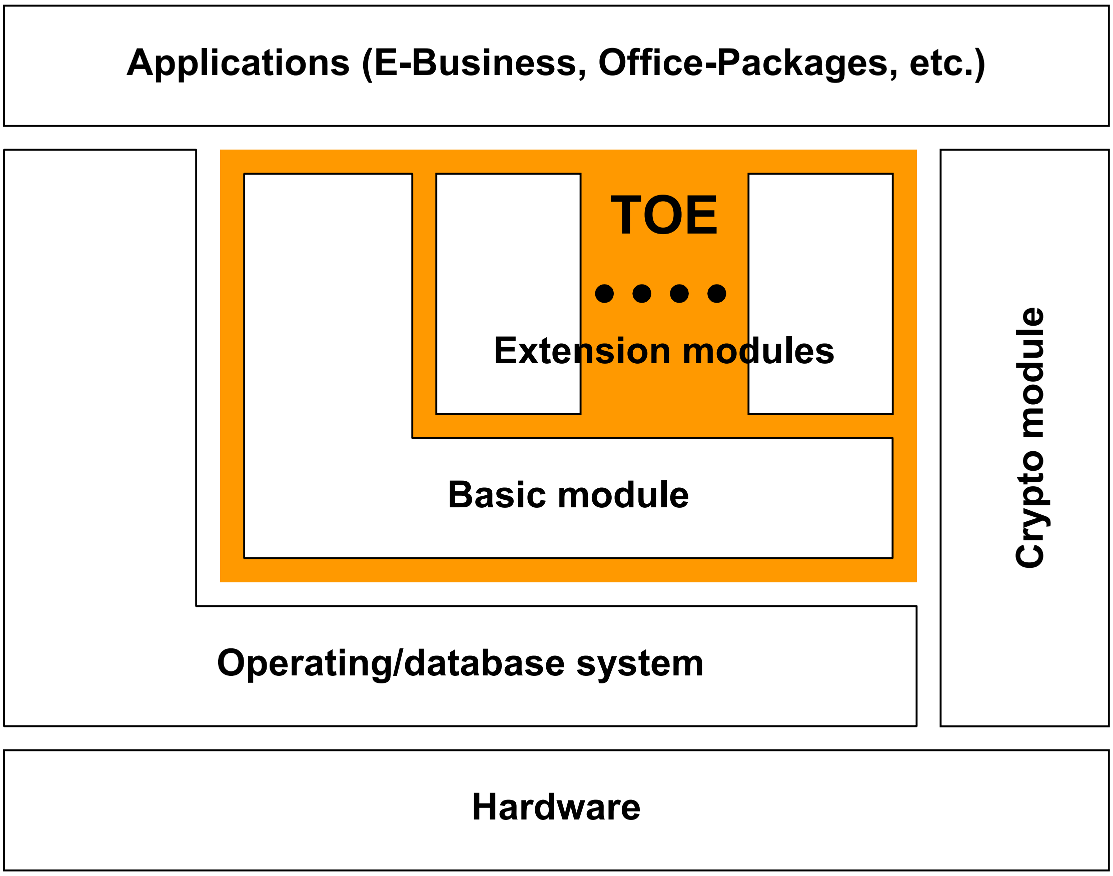

Applications (E-Business, Office-Packages, etc.)
TOE
● ● ● ●
Extension modules
Basic module
Operating/database system
Crypto module
Hardware

Figure 1: Possible structure of the TOE in the IT Environment

2.4 Operational Environment

It is assumed that every legitimate user of the IT system is interested in the security features of the TOE and that direct threats against user data do not originate from legitimate users. In order to be able to prevent undesired information flows, demanded by authorised users, a distinction of the users is not necessary.

It cannot be assumed that administrators of the IT system (in particular in the case of a remote administration) are equally interested in the security features of the TOE. Administrators of the IT system are regarded, just like unauthorised users (potential attackers), as originators of threats. It is assumed that the offensive capability is limited to the ability to execute obvious penetration attacks.

The TOE administrator has a privileged status. He/she is regarded as trustworthy without restriction. Measures that control the activity of the TOE administrator are therefore not intended.

2.5 TOE Security Policy

This chapter explains the security policy with which the TOE must comply, i.e. the "*discretionary information flow control" security functional policy (SFP)*. To describe the security policy, active units (the subjects), passive units (objects), and informations are distinguished. Objects may contain information and are the targets of operations, carried out by subjects. Subjects, objects and information have security attributes assigned to them. The policy decisions of the discretionary information flow control SFP are based on these attributes.

2.5.1 Concept Definitions

An **information flow** is the input or output of an IT component from/to any data depository. It is characterised by the specification of a subject, an object and an operation.

An information flow is caused by a **subject** which is identified by an active functional unit, such as for example an application (a user program to which tasks on the operation system level can be assigned).

The information flowing from/to subjects form, together with the container where the information is stored, the controlled **object**. Typically, the container is a data file; but it can also be considered as a data record in a database system or as the memory location that is assigned to an e-mail address.

The **data depository** describes the place where an object is kept. A data depository might relate to a local memory location medium, a computer address or an addressee that is reachable via a network connection. For a memory medium, the data depository can consist of a path information with respect to a hierarchical directory structure. An e-mail address would be typical for network connections.

Traditional postal service illustrates the mutual differentiation of the notions information, object, and data depository. The information corresponds to the content of a letter. This, together with the envelope, constitutes the object. The P.O. Box, or letter box, where the letter is kept corresponds to the data depository.

Finally, two operations are defined:

read(S; I; O)	subject S reads information I from the object O
write(S; I; O)	subject S writes information I in the object O. Any information that might have been in O before gets erased.

Application note 2: Often a read/write operation consists of several other operations, for instance it might be necessary to open a file beforehand and to close it finally. In the discretionary information flow control SFP a read/write operation is considered as a indivisible (atomic) operation. This is necessary in order to avoid conflicts in case where information flow rules are changed. Thus we abstract away from the concrete reality in the IT environment. Therefore, the manufacturer of a PP conformal product has to ensure that the atomicity of the two operations read(…) and write(…) is met.

An **Information Flow Rule** consists of the following components:

- an operation
- a set of subjects
- a set of data depositories
- a control flag (abbr. CF)
- a trust flag (abbr. TF)
- a protocol flag (abbr. PF)
- a set of information flow instructions.

Possible operations are reading (read) and writing (write) of information.
An information flow to or from controlled objects is permitted only in case the triggering subject belongs to the set of subjects mentioned in the rules. This allows one to determine the subjects that are approved for the appropriate information processing. Wildcards are allowed to be used to be able to describe sets of subjects in a compact way.
All information flows with any mentioned data depository are carried out in accordance with the information flow instructions. Again, wildcards are allowed to be used to be able to describe a set of data depositories in a compact way.
The control flag CF is set to “True” in order to indicate that information flows with these data depositories are only permitted in accordance with given information flow rules. By doing so, the essential characteristic feature of information flow control, namely that information that needs to be controlled remains within the controlled area, can be realised.
The trust flag TF is set “True“ when the mentioned subjects are permitted to write the information stored at the mentioned data depository to other data depositories without maintaining the protection. With this it is possible to specify exceptions to the scope of the controlled area.
The protocol flag PF is set “True“ whenever all demands for information flows which are permitted or rejected according to this information flow rule are to be recorded.
Via appropriate information rule instructions the authenticity, integrity, respectively confidentiality of the information stored at the mentioned data depository can be protected. In the information flow instructions concerning the operation writing of information, the least that has to be specified is that the information shall be encrypted or signed (if necessary, together with an indication which procedure and key should be applied for encrypting or signing). Accordingly, the least that has to be specified for the operation reading of information is that the information shall be decrypted (if necessary, together with an indication which decryption procedure and key should be applied). Moreover, in the information flow instructions, it must be possible to specify that the

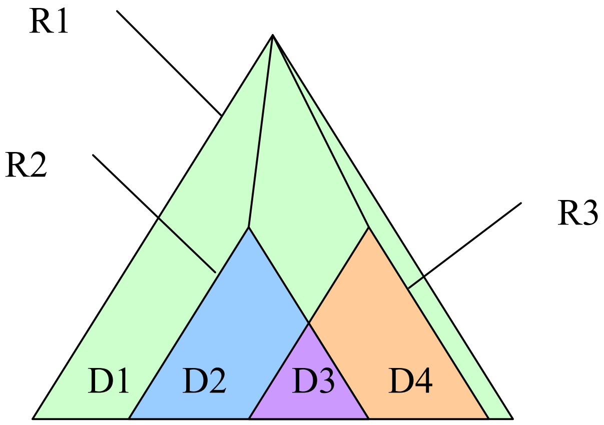
validity of used digital signatures has to be checked (if necessary, together with an indication which method and key should be applied).
In addition, the information flow instructions for reading and writing of information may contain some extra functions, for example, that it is indispensable to decompress the data under consideration before reading (possibly together with an indication on the de-compression program to be used). Or, that it has to be checked whether the data is virus-free (possibly together with an indication on the anti-virus program to be used) or whether the data has to be compressed before writing (possibly with an indication on the compression program to be used).

In order to meet different security needs, information flow rules can be combined in a **list of information flow rules**.

Given a list of information flow rules and a data depository D, an information flow rule R is said to be the **most specific information flow rule for D** if D is mentioned in the information flow rule R and there is no information flow rule R' in the list in which only some of the data depositories mentioned in R are mentioned.

With the notion of a most specific information flow rule it becomes possible to build a hierarchy of information flow rules. This allows one to identify those information flow rules whose focus is as close as possible to the considered data depository.

Illustration. As illustrated in Figure 2, information flow rule R1 mentions data depositories D1, D2, D3, and D4, rule R2 mentions deposits D2 and D3 and rule R3 mentions deposits D3 and D4. The table maps the given data depositories to the respective most specific information flow rule. It should be remarked that the sets of data depositories mentioned in the information flow rules R2 and R3 intersect. Since data depository D3 is mentioned in both rules, R2 as well as R3 are most specific information flow rules for D3.



	D1	D2	D3	D4
R1	×	—	—	—
R2	—	×	×	—
R3	—	—	×	×

Figure 2: Illustration of the notion "most specific information flow rule"

Overlapping of different information flow rules as in Figure 2 is particularly meaningful when wildcards for the designation of data depositories are used. In this case, general information flow rules can easily be formulated and, by the help of other rules, exceptions to these general rules can be defined. Obviously, conflicts between overlapping information flow rules are to be avoided or resolved. On the one hand, for each information flow requested by a concrete subject at most one of the most specific rules should be responsible. On the other hand, since different information flow rules may be applied depending on the requesting subject, the application of these rules to the same object should not lead to loss of data.

These considerations suggest suitable restrictions on the combination of information flow rules. A list of information flow rules is called **consistent** if the following conditions are met.

(C1) If the information flow instructions given in a information flow rule contains operations to ensure confidentiality, integrity, or authenticity, the control flag CF has to be set to “True”.

(C2) For each information flow there is at most one most specific information flow rule in which the subject in question and the current operation are mentioned.

(C3) If, for a data depository, there is a most specific information flow rule for the read operation, there is a most specific one for the write operation as well. If, for a data depository, there is a most specific information flow rule for the write operation, there is a most specific one for the read operation as well.

(C4) For any two most specific information flow rules mentioning the same data depository it holds that the information flow instructions given in the information flow rules do not contradict each other.

The ST author is bound to define what it means that two information flow instructions are contradictory. This definition has to be that precise, that an applicable procedure can be derived from it which allows one to determine whether information flow instructions are contradictory.

Explanation. If, for example, the information flow instructions of two information flow rules specify that different applications using different encryption procedures may write information in one and the same object, these information flow instructions contradict each other. On the other hand, it has to be considered that, for example, backup procedures should have reading access to any part of the storage media. The backup procedure is meant to read the data bit by bit and to write it to the backup medium bit by bit (i.e., without previous decryption). This way of processing must not contradict any other information flow instruction.

The discretionary information flow control to be carried through by the TOE is based on the object- and subject attributes specified in Table 1.

category	**attributes**	**possible values**
object (O)	control status (C(O))	Strong, Weak
subject (S)	security level (L(S))	High, Low

Table 1: Security attributes

The **control status** of an object serves to enforce the appropriateness and the protection of authenticity, integrity, respectively confidentiality of the information kept in this object. The value “Strong” indicates that the information in the referred object is to be protected.

The security attribute C(O) is invariable and statically assigned to the object O and therefore to the information therein. It takes the value ”Strong” if it is located at a data depository that is to be controlled. Otherwise, the security attribute C(O) takes the value “Weak”. In this context, a data depository D is said **to be controlled**, if the control flag CF in one of the most specific information flow rules for D has the value “True”.

Illustration. Assume that the list of information flow rules R1 up to R3 from Figure 2 is consistent. Let O be an object that is kept at data depository D3. The control status of O must not depend on whether the demanded information flow with O is controlled by rule R2 or R3. Otherwise a consistent assertion of a security policy for information flow control would not be possible. The definition of the security attribute C(O) determines how to cope with different statements in overlapping rules (R2 and R3 in the considered example). If, for instance, CF(R2) = “True” and CF(R3) = “False” then C(O) = “Strong”, independently from whether R2 or R3 is to be applied. To ease the administration of the TOE and to identify undesired effects in early stages, the plausibility check should give a hint in case the control flags of overlapping information flow rules are set to different values.

The **security level** of a subject serves to ensure the essential characteristic of information flow control, namely, that information which has to be controlled remains in the controlled area. The value “High” indicates that the concerned subject is in possession of information that is to be protected.
The safety attribute L(S) is variable and varies at run-time of the subject S. With the generation of a new subject S (e.g., when starting an application), the security attribute L(S) gets the value "Low" as an initial value. As soon as the subject S reads information from an object O with control status C(O) = “Strong”, the value of L(S) persistently changes to "High". There is one exception, namely information flows that are considered trustworthy, i.e., TF = “True” in the corresponding information flow rule.

2.5.2 Security Principles

The following security principles are defined for the given TOE security policy of discretionary information flow control.

(P1) **Recording**. Decisions on permission or rejection of information flows are recorded, provided this is required according to the information flow rules.

(P2) **Data Security.** Permitted information flows always take place in accordance with the information flow instructions mentioned in the information flow rules.

(P3) **Appropriation.** If the control status of an object O is “Strong” then information flows that are concerned with this object O are only permitted if they are requested from a subject S which is authorised to do so according to the information flow rules.

(P4) **Information Flow Control.** If the control status of an object O is “Strong” (C(O) = “Strong”) then an information I that originates from object O cannot be transferred into an object O’ with C(O’) = “Weak”, unless the triggering subject S is authorised to do so according to the information flow rules.

(P5) **Discretion.** As an exception to principle (P4 – Information Flow Control), at least the TOE administrator can explicitly authorise an information flow, i.e., an information I that originates from an object O with C(O) = “Strong” can be transferred into an object O’ with C(O’) = “Weak”.

Explanation. Principle (P4 – Information Flow Control) cannot be realised exclusively by using the security attribute control status, since the decision whether or not an information flow that writes into an object O’ is to be permitted depends on whether the information I originates from an object O with C(O) = “Strong”. To this end the security level of subject S is considered. In case where the information flow deals with the reading of information the attribute L(S) will be set to “High” if C(O) = “Strong”, unless the subject S is trustworthy, i.e. the trust flag TF has value “True” in the corresponding information flow rule.

Application note 3. The ST author may determine which users in addition to the TOE administrator are allowed to carry out authorisations according to principle (P5 – Discretion). In such a case he/she has to specify how these users are to be given the needed permission. For instance, this can be made with the help of ownership rights.

2.5.3 Security Characteristics

In what follows, the security characteristics of the discretionary information flow control SFP is given in detail.

Suppose a consistent list of information flow rules is given. In order to decide whether a demanded information flow is permitted or rejected, it has to be checked first whether the data depository belonging to the demanded information flow is mentioned in one of the information flow rules. If this is not the case, rules (CR1) respectively (CW1) are to be applied. Otherwise, before proceeding according to (CR2) or (CR3) respectively (CW2) or (CW3), it has to be checked first according to which of the existing information flow rules decisions have to be taken. In case the protocol flag of the information flow rules to be considered is set to “True” the decision will be recorded (viz. principle (P1 – Recording).

A **selection function**, which is a crucial parameter of the discretionary information flow control SFP, chooses the relevant information flow rule.

Let read(S; I; O) respectively write(S; I; O) be the demanded information flow and D the data depository where the object O is located. If the data depository D is mentioned in at least one of the information flow rules, the selection function chooses an information flow rule. The chosen information flow rule R has the following features:

(S1) Information flow rule R is a most specific information flow rule for data depository D in which the current operation is mentioned.[1]

(S2) If the list of information flow rules contains a most specific information flow rule for the data depository D mentioning both the current operation and the subject S, the subject S is mentioned as well in R.

[1] The consistency conditions specified in section Concept Definitions ensure that in a consistent list of information flow rules, for every data depository mentioned in an information flow rule there is a most specific information flow rule in which the current operation is mentioned.

The discretionary information flow control SFP relies on the rules that are listed in table 2. These rules are described in the following.

Legend *permitted/rejected:* Decision of the SFP (C…): Security Characteristics (P…): Security Principles		Choice function provides no rule	Choice function provides rule R		
			Object O has control status C(O) = "Weak"	Object O has control status C(O) = "Strong"	
				Subject S is mentioned in rule R	Subject S is not mentioned in rule R
read(S; I; O)		***permitted*** (CR1)	***permitted*** (CR2) (P1), (P2)	***permitted*** (CR3 (i)) (P1), (P2), (P3)	***rejected*** (CR3 (ii)) (P1), (P3)
write(S; I; O)	L(S) = "Low"	***permitted*** (CW1 (i)) (P4)	***permitted*** (CW2 (i)) (P1), (P2), (P4)	***permitted*** (CW3 (i)) (P1), (P2), (P3)	***rejected*** (CW3 (ii)) (P1), (P3)
	L(S) = "High"	***rejected*** (CW1 (ii)) (P4), (P5)	***rejected*** (CW2 (ii)) (P1), (P4), (P5)		

Table 2: Overview over the Security Characteristics

Reading of information Let read(S; I; O) be the demanded information flow and let D be the data depository where O is kept.

(CR1) If there is no information flow rule in which the data depository D is mentioned, the information flow is to be permitted. The value of the security attribute L(S) does not change.

(CR2) If there is a information flow rule in which the data depository D is mentioned and if the security attribute C(O) has the value "Weak", then the information flow is permitted. The read operation has to be carried out in accordance with the information flow instruction mentioned in the selected information flow rule.[2] The value of the security attribute L(S) does not change.

(CR3) If there is a information flow rule in which the data depository D is mentioned and if the security attribute C(O) has the value "Strong", two cases are distinguished.

(i) The information flow is permitted if the subject S is mentioned in the selected information flow rule R. The read operation has to be carried out in accordance with the information flow instruction mentioned in R.[2] If the trust flag in R is set to "False", the value of the security attribute L(S) is set to "High", otherwise the value of the security attribute L(S) does not change.

(ii) If the subject S is not mentioned in the information flow rule R, the information flow is to be rejected and the user is to be informed about that. The value of the security attribute L(S) does not change.

[2] The use of consistent lists of control rules ensures that the operation Read is mentioned in the selected information flow rule.

Writing of information Let write(S; I; O) be the demanded information flow and let D be the data depository where O is kept.

(CW1) If there is no information flow rule in which the data depository D is mentioned, the following two cases are distinguished:

(i) If the security attribute L(S) has the value “Low”, the information flow is to be permitted. The value of the security attribute L(S) does not change.

(ii) If the security attribute L(S) has the value “High”, the information flow is to be rejected and the user is to be informed about it. The value of the security attribute L(S) does not change.

(CW2) If there is a information flow rule in which the data depository D is mentioned and if the security attribute C(O) has the value “Weak”, the following two cases are distinguished:

(i) If the security attribute L(S) has the value “Low”, the information flow is to be permitted. The write operation has to be carried out in accordance with the information flow instruction mentioned in the selected information flow rule.[3] The value of the security attribute L(S) does not change.

(ii) If the security attribute L(S) has the value “High”, the information flow is to be rejected and the user is to be informed about it. The value of the security attribute L(S) does not change.

(CW3) If there is a information flow rule in which the data depository D is mentioned and if the security attribute C(O) has the value “Strong”, the following two cases are distinguished:

(i) If the subject S is mentioned in the selected information flow rule R, the information flow is to be permitted. The write operation has to be carried out in accordance with the information flow instruction given in the information flow rule R.[3] The value of the security attribute L(S) does not change.

(ii) If the subject S is not mentioned in the selected information flow rule R, the information flow is to be rejected and the user is to be informed about it. The value of the security attribute L(S) does not change.

Information flows which are to be rejected according to rules (CW1 (ii) and (CW2 (ii)) may be permitted if these information flows are explicitly authorised (see principle (P5 – Discretion)). Such explicit authorisations are to be recorded.

[3] The use of consistent lists of information flow rules ensures that the operation Write is mentioned in the selected information flow rule.

3 TOE Security Environment

3.1 Description of roles and assets

3.1.1 Roles

The TOE recognises the following roles:

TOE-Administrator A TOE-Administrator is in charge of the security-specific configuration and de-activation of the TOE. In particular, it is the TOE-Administrator's task to define the information flow rules and to evaluate the protocols.

IT-Administrator An IT-Administrator installs the TOE and maintains the IT system, except for the TOE.

IT-User An IT-User uses the IT system as usual.

Users who are not allowed to act in one of the roles mentioned here are referred to as unauthorised users below.

Application note 4. The TOE shall differentiate between the roles of IT-User and IT-Administrator. For this purpose, the TOE may distinguish two modes, a maintenance mode and a standard mode. Each user, having access to the TOE in the standard mode, acts in the role IT-User, whereas a user, having access to the TOE in the maintenance mode, acts in the role IT-Administrator.

Application note 5. The role TOE-Administrator can be split into several roles, for example, in order to spread the determination of information flow rules into several responsibilities. The ST author has to ensure the consistency of the entire list of stated information flow rules. A possible approach is the allocation of disjoint responsibilities or the establishment of hierarchical relations between the different variants of the role TOE-Administrator.

Application note 6. The ST author may introduce further roles, e.g. a TOE-Auditor, to establish the auditing capability of the TOE. A TOE-Auditor should be able to review, archive, and clear protocol data. In this case a TOE-Administrator should only have the authorisation to evaluate protocol data concerning permitted and rejected information flows.

3.1.2 Assets

As for the assets to be protected, one distinguishes between primary assets and secondary assets. Primary assets are assets whose protection is the actual task of the TOE. It is true that secondary assets are worth being protected, but they would not exist without the TOE. If the secondary assets are not protected, the TOE is unable to protect the primary assets.

3.1.2.1 Primary Assets

UserData The UserData are those data processed by the IT-User within the scope of his/hers activity. They are to be protected within the IT system as well as during transfer.

3.1.2.2 Secondary Assets

TSF Data TSF data are:

ProtocolData The ProtocolData include all events recorded by the TOE. They contain in particular records concerning permitted/rejected information flows.

RuleData The RuleData include the list of defined information flow rules. An information flow rule contains the details described in section 2.5.1.

Explanation. The existence of the TOE triggers additional assets to be protected, such as e.g. the RuleData. Furthermore, the functioning of the TOE is to be checkable. Therefore, permitted and rejected information flows are to be auditable. This way, a TOE-Administrator has an opportunity to check whether the TOE behaves as intended.

TSF data, that are not ProtocolData respectively RuleData are referred to as other TSF data below.

Application note 7. In case of a composite TOE, there may be additional assets that have to be protected, e.g., keys for crypotographic operations. In this case, there may be additional threats against these assets. The ST author has to take these aspects into consideration.

3.2 Assumptions

A.NoBypass The TOE is integrated in the IT environment in a way that all information flows to be protected are passed through the TOE.

A.Selection The IT environment provides the TOE with reliable time stamps and correct information for the identification of required information flows, i.e. subject identity, operation and data depository.

A.Qualification An IT-Administrator and a TOE-Administrator possess an appropriate qualification.

- A TOE-Administrator has the ability to administer the TOE. In particular, he/she has the ability to define information flow rules and to evaluate protocol data. He/she handles the information contained in the recorded protocols confidentially.
- An IT-Administrator has the ability to install the TOE.

A.NoCapture Running sessions of legitimate users cannot be taken over by persons that are not allowed to act in this role (IT-administrator, unauthorised users).

Explanation. The assumption A.NoCapture guarantees that a session once started by an IT-User cannot be continued by unauthorised persons (potential attackers). In order to achieve this within as well as outside the regular operating times of the IT environment, one has to appropriately employ given security mechanisms (e.g. locking of the session or shutdown of the IT system). Within the scope of this protection profile no exact specification of the security mechanisms that are to be provided is given (e.g. by a choice of functional components from part 2 of the CC that are to be provided by the IT environment). The manufacturer has to show in which way the assumption A.NoCapture is maintained by the IT environment. The protection of the sessions of a TOE-Administrator is ensured by the TOE (see O.Impersonate). It is pointed out that the correctness of the role assignment is the TOE's task and is not guaranteed by the assumption A.NoCapture.

A.NoVirus Malicious software is not part of controlled subjects.

Explanation. In practice, the absence of malicious software cannot be guaranteed in any available IT environment. A TOE that is conformant with this protection profile reduces the impact of malicious software. As any software, malicious software can only act within the boundaries of the information flow control. The assumption A.NoVirus is necessary since it is not part of the security features of the TOE to evaluate how trustworthy the subjects are.

3.3 Threats

Below, the basic features of the TOE are compared with the identified threats. This is to show clearly which features add to the protection against which kind of threat. In addition, each mentioned threat is assigned to its threat agent. The following abbreviations are used in Table 3: IT-A(dministrator), IT-U(ser), TOE-A(dministrator), Un(authorised) U(ser), T.Info(rmationFlow), T.Confi(dentiality), T.Mani(pulate), T.Unaw(are), T.Imp(ersonate) , T.Sup(port) and T.Modi(fication).

Features	**Threat Agent**			
	IT-A	**IT-U**	**TOE-A**	**UnU**
Rejection of information flows: information flow control, appropriation	T.Spy T.Write T.Imp	T.Info T.Imp	—	T.Spy T.Write T.Imp
Protection of information flows taking place: confidentiality, integrity, authenticity	T.Confi	T.Info T.Confi	—	T.Read T.Mani T.Confi
Automated and transparent usage of security functions	—	T.Info T.Unaw	—	T.Read T.Mani
Support with the administration	T.Modi	T.Modi	T.Sup	T.Modi
Recording of information flows having taken place and of those having been rejected	T.Spy T.Write	T.Info	—	T.Spy T.Write T.Read T.Mani

Table 3: Assignment of security features to threats

Application note 8. The assignment of security features to threats is rather informative. It is meant to illustrate the functionality of the TOE. The ST author's focus is on the threats (as well as the assumptions and the organisational security policies) defined below. These are crucial for the definition of the security objectives and IT security requirements.

3.3.1 Threat agents

By doing mistakes, a TOE-Administrator, an IT-Administrator and an IT-User can be a threat agent. In addition, threats can be triggered by an IT-Administrator and an unauthorised user, possibly by using so called malicious software (e.g. viruses, Trojan Horses).

3.3.2 Primary threats

T.InformationFlow An IT-User triggers information flows making it possible for unauthorised users to spy out respectively to manipulate UserData that are to be protected regarding confidentiality, integrity respectively authenticity.

Explanation. Without the TOE an IT-User might for example inadvertently cause confidential data to be transferred unencrypted via open nets.

T.Read UserData worth being protected regarding confidentiality is read by an unauthorised user during a proceeding information flow.

Explanation. This threat aims in particular at the transfer of data via open nets. First and foremost the focus here is on the wide spread custom to sent data like for example credit card or patient data unencrypted. An unauthorised user might spy out such a communication and thus get a hold of sensitive UserData.

T.Spy An IT-Administrator or an unauthorised user triggers an information flow (possibly using malicious software) in order to spy out UserData worth being protected with regard to confidentiality.

Explanation. This threat is to be understood in analogy to the threat T.Read. The difference is that the aim here is not to spy out unprotected information channels but to read stored UserData (e.g. on hard disks or floppy disks).

T.Manipulate UserData worth being protected regarding integrity respectively authenticity are manipulated by an unauthorised user during a proceeding information flow, without being noticed.

Explanation. This threat aims mainly at the integrity and authenticity of UserData, not at their confidentiality. UserData should be protected against unauthorised manipulation, even if they are not confidential.

T.Write An IT-Administrator or an unauthorised user triggers an information flow (possibly by using malicious software) in order to manipulate UserData worth being protected regarding integrity respectively authenticity, without being noticed.

Explanation. Analogous to threat T.Manipulate.

T.Unaware Available mechanisms to protect the integrity, authenticity and confidentiality of UserData are used insufficiently or not at all through ignorance respectively negligence of an IT-User.

Explanation. This threat describes the situation where an IT-User has the possibility to protect his/her UserData against unauthorised access, but does not do so – may it be through ignorance or because the use of protection mechanisms may seem too troublesome or too complicated.

3.3.3 Secondary threats

T.Modification An IT-Administrator, an IT-User or an unauthorised user modifies (possibly by using malicious software) the TSF data, in a way that:

- specified security policies are evaded.
- the integrity and confidentiality of UserData are lost.

Explanation. An IT-Administrator, an IT-User, and an unauthorised user (possibly by using malicious software) might for example change the RuleData in a way that a security policy, specified by a TOE-Administrator, cannot be realised any more.

T.Confidentiality An IT-Administrator, an IT-User or an unauthorised user takes note of the ProtocolData.

Explanation. The unauthorised knowledge of ProtocolData contradicts basic principles of privacy protection.

T.Impersonate An unauthorised person acts in the role IT-User respectively TOE-Administrator.

Explanation. Like an IT-user, an IT-administrator or an unauthorised person can trigger information flows with controlled objects. A person who is not allowed to administer the TOE (an IT-user, an IT-administrator or an unauthorised user) receives the privileges of the role TOE-administrator and can later on shutdown the transparently working TOE, modify RuleData or take note of ProtocolData.

T.Support The TOE is administered incorrectly through a TOE-Administrator's ignorance respectively negligence.

Explanation. As a consequence of administration mistakes, a TOE-Administrator might specify RuleData in a way such that the integrity and confidentiality of the User-Data are lost.

3.4 Organisational Security Policies

P.Appropriation It must be possible to specify which subjects (e.g. applications) are allowed to process UserData.

Explanation. This organisational security policy derives from the following principle of appropriateness of privacy protection:

> Data may be collected, processed or used exclusively for the purpose it was destined for. It has to be laid down in detail who is allowed to exploit the data, under what circumstances, using which means, in which period of time and to which purpose.

4 Security Objectives

In the beginning of this chapter, the security objectives are assigned to the features of the TOE for the realisation of which they are crucial.

Features	Security Objectives
Rejection of information flows: Information flow control, Appropriation	O.InformationFlow, O.Impersonate, O.Support, O.TOE-Administration
Protection of proceeding information flows: confidentiality, integrity, authenticity	O.InformationFlow, O.Disclosure, O.Manipulation, O.Support, O.Impersonate, OE.Disclosure, OE.Manipulation
Automated and transparent usage of security functions	O.InformationFlow, O.Support, O.TOE-Administration
Support with the administration	O.Support, O.TOE-Administration
Recording of permitted and rejected information flows	O.Support, O.TOE-Administration

Table 4: Assignment of features to security objectives

Application note 9. The assignment of features to security objectives has, like their assignment to threats, informative character. It is meant to illustrate the functionality of the TOE. The ST author's focus is on the security objectives defined below. These are crucial for the definition of the IT security requirements.

4.1 Security objectives for the TOE

O.InformationFlow The TOE controls information flows according to the specified TOE security policy. For this the TOE guarantees that

- information that has to be protected with respect to confidentiality, integrity respectively authenticity can only leave the controlled area if this keeps to the TOE security policy.
- UserData which are subject to the principle of appropriateness may be processed exclusively by authorised subjects.
- information flows with controlled data depositories may be triggered exclusively by a TOE-Administrator and an IT-User.

Explanation. The TOE has to provide means to specify information flow rules. The accomplishment of these information flow rules is to be enforced. In order to do so, it is necessary for the TOE to be able to identify subjects and objects.

Application note 10. The formulation of O.InformationFlow does not specify the considered information flows in detail. The ST author has put this in concrete form. This way, the producer gains a lot of flexibility concerning the operational area and the construction of the TOE.

O.Disclosure The protection of UserData against unauthorised access within the IT system, as well as during the transfer is enforced by the use of encryption procedures.

O.Manipulation The use of signature procedures ensures that the manipulation of UserData within the IT system or during transmission does not remain unnoticed.

Explanation. The formulation of the security objectives O.Disclosure and O.Manipulation is identical to the corresponding security objectives for the environment. This is necessary according to the Common Criteria[4], because both security objectives are assigned partly to the TOE and partly to the environment: While the choice of the information flow instructions is a task for the TOE, the cryptographic operations that are formulated in the information flow instructions are executed in its environment.

O.Support TOE supports the activity of a TOE-Administrator by

- indicating possible conflicts with existing information flow rules, when creating a new information flow rule.
- providing means (e.g. wildcards) which simplify the definition of information flow rules.
- making all permitted and rejected information flows recordable and thus providing information for the validation of the set of stated information flow rules.

Explanation. The quality of the security features of the TOE is mainly determined by the appropriateness of the employed information flow rules. It is thus substantial to support the generation and the validation of RuleData.

[4] Quotation from the Common Criteria, Part 1, Section B.2.5: “Note: when a threat or organisational security policy is to be covered partly by the TOE and partly by its environment, then the related objective shall be repeated in each category.”

O.TOE-Administration The TOE ensures that

- exclusively a TOE-Administrator has the right to change the RuleData.
- the TOE, once started and installed, may be deactivated exclusively by a TOE-Administrator.
- exclusively a TOE-Administrator has access to the ProtocolData.

Explanation. Only persons with the required competence may specify respectively change the RuleData. As the TOE works transparently, it has to be ensured that the TOE in fact provides the expected security services. The ProtocolData contain details about the activities of the individual IT-Users, that must not be freely accessible for reasons of privacy protection. A TOE-Administrator has to have access to the ProtocolData, as he/she has to use them in order to be able to judge the quality of the specified information flow rules.

O.Impersonate The TOE ensures that

- only authorised persons can act in the role TOE-administrator.
- the roles IT-user and IT-administrator are correctly assigned.
- the assignment of the role IT-user has to be authorized by the IT-administrator and, in addition, it is impossible to act in this role beforehand.
- the role IT-administrator and one of the roles IT-user and TOE-administrator cannot be act simultaneously.

Explanation. The trustworthiness of the TOE-administrator's activities requires a legitimation by the TOE as a basis for the assignment of roles. In addition, the TOE has to ensure that the privileges of the role TOE-administrator cannot be taken over by an unauthorised person during a TOE-administration session. Since it is not required, that the IT environment supports the distinction of users, the assignment of the role IT-user requires an explicit authorization by the TOE-administrator. Also, it is to be assumed that an IT-administrator (in particular in case of a remote administration of the IT-system) is not interested in the security service of the TOE. Therefore, it should be impossible to act in the role IT-user or in the role TOE-administrator during an IT-administration.

4.2 Security objectives for the environment

OE.Disclosure The protection of UserData against unauthorised access within the IT system as well as during the transfer is ensured by the use of encryption procedures.

OE.Manipulation The use of signature procedures ensures that a manipulation of UserData within the IT system as well as during the transmission does not remain unnoticed.

Application note 11. If the cryptographic support is integrated in the TOE (see also chapter PP Application Notes), then the security objectives OE.Disclosure and OE.Manipulation can be dropped, since they are covered by the corresponding objectives for the TOE.

OE.NoBypass The TOE is integrated in the IT environment in a way that all information flows to be protected are passed through the TOE.

OE.Selection The TOE is provided reliable time stamps and correct information for the identification of demanded information flows i.e. subject identity, operation and data depository.

OE.Qualification A TOE-Administrator and an IT-Administrator have an appropriate qualification.

- A TOE-Administrator has the qualification to administer the TOE. In particular he/she has the qualification to define information flow rules and to evaluate protocol data. He/she treats the information contained in the recorded audits confidentially.
- An IT-Administrator has the qualification to install the TOE.

Explanation. Some crucial aspects are pointed out here: a TOE-Administrator has to know how to specify the RuleData, so the UserData can be protected appropriately (e.g. in accordance with legal regulations). In addition, a TOE-Administrator has to be able to evaluate the ProtocolData.

OE.NoCapture Running sessions of legitimate users cannot be taken over by persons that are not allowed to act in this role (IT-administrator, unauthorised users).

OE.NoVirus Malicious software does not go into action as a part of controlled subjects.

Application note 12. Maybe the TOE can contribute to the enforcement of the security objective OE.NoVirus. For instance, suitable measures might be taken to prevent from undesired information flows in case of noticeable activities of malicious software (see O.InformationFlow). In such a case the ST author is urged to check whether OE.NoVirus has to be repeated in subsection Security Objectives for the TOE.

5 IT security requirements

This section comprises functional requirements that shall be fulfilled by a product (and its IT environment) that is conformant to this protection profile. The requirements consist of functional components of part 2 of the CC. Table 5 relates the functional classes of the CC to the security features of the TOE.

Security Features	Functional Classes (according to CC)
Rejection of information flows: information flow control, appropriateness	FDP (User data protection), FIA (Identification and authentication) FTA (TOE access)
Protection of proceeding information flows: confidentiality, integrity, authenticity	FCS (Cryptographic support), FDP (User data protection), FIA (Identification and authentication), FPT (Protection of the TSF) FTA (TOE access)
Transparent and automatic application of security functions	FMT (Security management)
Administration support	FAU (Security audit), FMT (Security management)
Auditing of permitted and rejected information flows	FAU (Security audit)

Table 5: Assignment between security features and CC functional classes

5.1 TOE security requirements

Minimum strength of function level

For the TOE security functions that are realised by a probabilistic or permutational mechanism the minimum strength level SOF-medium is postulated.

5.1.1 TOE security functional requirements

Table 6 provides an overview of the security functional requirements that shall be fulfilled by the TOE.

No.	Component	Description
1.	FAU_GEN.1	Audit data generation
2.	FAU_SAR.1	Audit review
3.	FAU_SAR.2	Restricted audit
4.	FAU_SAR.3	Selectable audit review
5.	FAU_SEL.1	Security audit event selection
6.	FAU_STG.1	Protected audit trail storage
7.	FAU_STG.3	Actions in case of possible audit data loss
8.	FDP_ETC.1	Export of user data without security attributes
9.	FDP_IFC.1	Subset information flow control
10.	FDP_IFF.1	Simple security attributes
11.	FDP_ITC.1	Import of user data without security attributes
12.	FIA_UAU.1	Timing of authentication
13.	FIA_UID.1	Timing of identification
14.	FIA_UID.2	User identification before any action
15.	FMT_MOF.1	Management of security functions behaviour
16.	FMT_MSA.1	Management of security attributes
17.	FMT_MSA.3	Static attribute initialisation
18a.	FMT_MTD.1A	Management of TSF-Data
18b.	FMT_MTD.1B	
19.	FMT_MTD.3	Secure TSF data
20.	FMT_SMF.1	Specification of Management Functions
21.	FMT_SMR.2	Restriction of security roles
22.	FTA_SSL.3	TSF-initiated termination

Table 6: TOE security functional requirements

5.1.1.1 Class FAU: Security audit

FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

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

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

b) All auditable events for the *minimum*[5] level of audit; and

c) Decisions to reject demanded information flows (FDP_IFF.1)[6].

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, *the additional audit relevant information listed in table 7*[7].

Dependencies: FPT_STM.1 Reliable time stamps

The auditable events for the selected audit level are summarised in table 7.

[5] [selection: minimum, basic, detailed, not specified]

[6] [assignment: other specifically defined auditable events]

[7] [assignment: other audit relevant information]

CC Component	Auditable event	Additional audit relevant information
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.	
FDP_ETC.1	Successful export of information.	
FDP_IFF.1	Decisions to permit requested information flows.	Selected information flow rules, security attributes, explicit authorisation
FDP_ITC.1	Successful import of user data, including any security attributes.	
FIA_UAU.1	Unsuccessful use of the authentication mechanism.	
FIA_UID.1	Unsuccessful use of the user identification mechanism, including the user identity provided.	
FIA_UID.2	Unsuccessful use of the user identification mechanism, including the user identity provided.	
FMT_MTD.3	All rejected values of TSF data.	
FMT_SMF.1	Use of the management functions.[8]	
FMT_SMR.2	Modifications to the group of users that are part of a role; Unsuccessful attempts to use a role due to the given conditions on the roles.	
FTA_SSL.3	Termination of an interactive session by the session locking mechanism.	

Table 7: Events for the audit level „minimum“

[8] Inserted according to the CC *Final Interpretation 065*.

FAU_SAR.1 Audit review

Hierarchical to: No other components.

FAU_SAR.1.1 The TSF shall provide *the role TOE-Administrator*[9] with the capability to read *permitted and rejected information flows*[10] 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.

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except *the role TOE-Administrator*[11].

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.3 Selectable audit review

Hierarchical to: No other components.

FAU_SAR.3.1 The TSF shall provide the ability to perform *searches and sorting*[12] of audit data based on *data depositories, subjects, time periods, information flow rules, and security attributes*[13].

Dependencies: FAU_SAR.1 Audit review

[9] [assignment: authorised users]
[10] [assignment: list of audit information]
[11] [refinement: those users that have been granted explicit read-access]
[12] [selection: searches, sorting, ordering]
[13] [assignment: criteria with logical relations]

FAU_SEL.1 Selective audit

Hierarchical to: No other components.

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) *object identity, subject identity, and event type*[14]

b) *protocol flag of the information flow rule selected by the selection function*[15]

Dependencies: FAU_GEN.1 Audit data generation
FMT_MTD.1 Management of TSF data

FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

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

FAU_STG.1.2 The TSF shall be able to [selection: *prevent, detect*] modifications to the audit records.

Dependencies: FAU_GEN.1 Audit data generation.

FAU_STG.3 Action in case of possible audit data loss

Hierarchical to : No other components.

FAU_STG.3.1 The TSF shall [assignment: *actions to be taken in case of possible audit storage failure*], if the audit trail exceeds [assignment: *pre-defined limit*].

Dependencies: FAU_STG.1 Protected audit trail storage

[14] [selection: object identity, user identity, subject identity, host identity, event type]

[15] [assignment: list of additional attributes that audit selectivity is based upon]

5.1.1.2 Class FDP: User data protection

FDP_ETC.1 Export of user data without security attributes

Hierarchical to: No other components.

FDP_ETC.1.1 The TSF shall enforce the *discretionary information flow control SFP*[16] when exporting user data, controlled under the SFP(s), outside of the TSC.

FDP_ETC.1.2 The TSF shall export the user data without the user data´s associated security attributes.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_IFC.1 Subset information flow control

Hierarchical to: No other components.

FDP_IFC.1.1 The TSF shall enforce the *discretionary information flow control SFP*[17] on *all subjects, objects out of which the information can be read or rather in which the information can be written, and the operation reading and writing of information*[18].

Dependencies: FDP_IFF.1 Simple security attributes

Explanation. A description of the discretionary information flow control SFP and of the operations reading and writing of information is given in section 2.5.

Application note 13. The choice of component FDP_IFC.1 represents a minimal requirement which gives the manufacturer as much flexibility as possible (see Application note 9). If necessary, the ST author can replace FDP_IFC.1 by the component FDP_IFC.2 (Complete information flow control) which is hierarchically above.

[16] [assignment: access control SFP(s) and/or information flow control SFP(s)]

[17] [assignment: information flow control SFP]

[18] [assignment: list of subjects, information, and operations that cause controlled information to flow to and from subjects covered by the SFP]

FDP_IFF.1 Simple security attributes

Hierarchical to: No other components.

FDP_IFF.1.1 The TSF shall enforce the *discretionary information flow control SFP*[19] based on the following types of subject and information security attributes*: Security level of the subject and control level of the object out of which the information can be read or in which the information can be written*[20]*.*

FDP_IFF.1.2 The TSP shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold[21]: *The appropriate security characteristics for the enforcement of the security principles (P1), (P2), (P3) and (P4) are:*

a) For the operation reading of information: The information flow shall be permitted when the requirements expressed in the rules (CR1), (CR2) or (CR3 (i)) are met.

b) For the operation writing of information: The information flow shall be permitted when the requirements expressed in the rules (CW1 (i)), (CW2 (i)), or (CW3 (i)) are met.

FDP_IFF.1.3 The TSF shall enforce *no additional information flow control SFP rules*[22].

[19] [assignment: information flow control SFP]

[20] [assignment: the minimum number and type of security attributes]

[21] [assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes]

[22] [assignment: additional information flow control SFP rules]

FDP_IFF.1.4 The TSF shall provide the following *list of additional SFP capabilities:*[23]

a) A selection function that selects only information flow rules with the features (S1) and (S2).

b) In the information flow instructions that are components of the information flow rules there must be at least specifiable that

i. encryption procedures according to FCS_COP.1A, FCS_COP.1B, and FCS_COP.1C that are provided by the IT environment are used.

ii. procedure for generating and testing electronic signatures according to FCS_COP.1D, FCS_COP.1E, FCS_COP.1F1C that are provided by the IT environment are used.

c) For the read operation: When the requirements expressed in the rules (CR2) or (CR3 (i)) are met, the operation Read shall take place according to the information flow instruction mentioned in the selected information flow rule.

d) For the write operation: When the requirements expressed in the rules (CW2) or (CW3) are met, the operation Write shall take place according to the information flow instruction mentioned in the selected information flow rule.

e) The decision on auditing the demand of an information flow has to be recorded if the protocol flag of the selected information flow rule is set „True“ or if the information flow is explicitly authorised according to FDP_IFF.1.5 (b).

[23] [assignment: list of additional SFP capabilities]

FDP_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules[24]:

a) All information flows with the monitor and the keyboard are permitted.

b) Information flows are to be permitted if the conditions for the rules (CW1 (ii)) or (CW2 (ii)) are fulfilled and if they are explicitly authorised by a user (at least TOE-Administrator) who is authorised to do so (security principle (P5)).

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules[25].

a) All information flows with objects whose security attribute control status has the value "Strong" shall be rejected, if these information flows are not demanded by one of the roles TOE-Administrator or IT-User.

Dependencies:	FDP_IFC.1	Subset information flow control
	FMT_MSA.3	Static attribute initialisation

Explanation. A description of the discretionary information flow control SFP including the quoted security principles and characteristics can be found in section 2.5.

[24] [assignment: rules, based on security attributes, that explicitly authorise information flows]

[25] [assignment: rules, based on security attributes, that explicitly deny information flows]

FDP_ITC.1 Import of user data without security attributes

Hierarchical to: no other components.

FDP_ITC.1.1 The TSF shall enforce the *discretionary information flow control SFP*[26] when importing user data, controlled under the SFP, from outside of the TSC.

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

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: *none*[27].

Dependencies:	[FDP_ACC.1	Subset access control, or
	FDP_IFC.1	Subset information flow control]
	FMT_MSA.3	Static attribute initialisation

Explanation. A description of the discretionary information flow control SFP is given in section 2.5.

[26] [assignment: access control SFP and/or information flow control SFP]

[27] [assignment: additional importation control rules]

5.1.1.3 Class FIA: Identification and authentication

The components of this class settle the (authorised) user's permission to execute TSF-mediated actions. In particular the actions that enforce the principles (P1) to (P5) of the SFP for the discretionary information flow control are concerned. These actions correspond to the regular usage of the TOE: The user demands information flows and the TSF execute actions that control these information flows to enforce the security principles (P1) to (P5).
In order to guarantee the aimed transparency, authorised users should be able to use the IT-system as usual. Therefore, the control of the information flows has to be performed on the basis of an identification and a role assignment (see FMT_SMR.2) that do not demand for an independent authentication by the TOE. This transparency is not in force, however, when cryptographic keys are issued; in particular, if the cryptographic support is sourced out to other products.
Based on an explicit authorization by the TOE-administrator, the TOE assigns the roles IT-user or IT-administrator to the users (see FMT_SMR.2). The component FIA_UID.2 below guarantees that this role assignment by the TOE precedes the execution of TSF-mediated actions. Furthermore, the components FIA_UAU.1 and FIA_UID.1 ensure that actions different from those that enforce the principles (P1) to (P5) of the SFP of the discretionary information flow control are only allowed for a (by the TOE) successfully identified and authenticated TOE-administrator.

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.

FIA_UAU.1.1 The TSF shall allow *actions for the enforcement of the principles (P1) - (P5) of the discretionary information flow control SFP*[28] on behalf of the user to be performed before the user is authenticated *as TOE-Administrator*[29].

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated *as TOE-Administrator*[30] before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: FIA_UID.1 Timing of identification

[28] [assignment: list of TSF-mediated actions]
[29] [refinement]
[30] [refinement]

FIA_UID.1 Timing of identification

Hierarchical to: No other components.

FIA_UID.1.1 The TSF shall allow *actions for the enforcement of the principles (P1) - (P5) of the discretionary information flow control SFP*[31] on behalf of the user to be performed before the user is identified *as TOE-Administrator*[32].

FIA_UID.1.2 The TSF shall require each user to be successfully identified *as TOE-Administrator*[33] before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: No dependencies

FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1

FIA_UID.2.1 The TSF shall require each user to identify itself *as IT-User or IT-Administrator*[34] before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: No dependencies

[31] [assignment: list of TSF-mediated actions]
[32] [refinement]
[33] [refinement]
[34] [refinement]

5.1.1.4 Class FMT: Security management

FMT_MOF.1 Management of security functions behaviour

Hierarchical to: No other components.

FMT_MOF.1.1 The TSF shall restrict the ability to *disable*[35] the functions *for the implementation of the discretionary information flow control SFP*[36] to *the TOE-Administrator*[37].

Dependencies: FMT_SMR.1 Security roles

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the *discretionary information flow control SFP*[38] to restrict the ability to *change_default*[39] the security attributes *control status of an object*[40] to *the TOE-Administrator*[41].

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow]
FMT_SMF.1 Specification of Management Functions[42]
FMT_SMR.1 Security roles

[35] [selection: determine the behaviour of, disable, enable, modify the behaviour of]
[36] [assignment: list of functions]
[37] [assignment: the authorised identified roles]
[38] [assignment: access control SFP, information flow control SFP]
[39] [selection: change_default, query, modify, delete, [assignment: other operations]]
[40] [assignment: list of security attributes]
[41] [assignment: the authorised identified roles]
[42] Included according to the CC *Final Interpretation 065*.

FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the *discretionary information flow control SFP*[43] to provide *permissive*[44] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow *no role*[45] to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

This component is refined by the following element.

FMT_MSA.3.a Provision of permissive default values means for the security attribute control status that the value of this attribute shall be set to “Weak” if the object is kept or generated at a data depository that is not controlled.

FMT_MSA.3.b Provision of permissive default values means for the security attribute security level that the value of this attribute shall be set to “Low” when generating a new subject (e.g. by starting an application).

[43] [assignment: access control SFP, information flow control SFP]
[44] [selection: restrictive, permissive, other property]
[45] [assignment: the authorised identified roles]

FMT_MTD.1A Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1A.1 The TSF shall restrict the ability to *modify, delete and add*[46] the *RuleData and other TSF data*[47] to *the TOE-Administrator*[48].

Dependencies: FMT_SMF.1 Specification of Management Functions[49]
FMT_SMR.1 Security roles

FMT_MTD.1B Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1B.1 The TSF shall restrict the ability to *query and clear*[50] the *ProtocolData*[51] to *the TOE-Administrator*[52].

Dependencies: FMT_SMF.1 Specification of Management Functions[53]
FMT_SMR.1 Security roles

[46] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[47] [assignment: list of TSF data]
[48] [assignment: the authorised identified roles]
[49] Included according to the CC *Final Interpretation 065*.
[50] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[51] [assignment: list of TSF data]
[52] [assignment: the authorised identified roles]
[53] Included according to the CC *Final Interpretation 065*.

FMT_MTD.3 Secure TSF data

Hierarchical to: No other components.

FMT_MTD.3.1 The TSF shall ensure that only secure values are accepted for TSF data.

The component is refined through the following element.

FMT_MTD.3.a The TSF shall ensure that only consistent lists of information flow rules, i.e. lists that meet the conditions (C1) to (C4), are accepted.

Dependencies: ADV_SPM.1 Informal TOE security policy model
FMT_MTD.1 Management of TSF data

Explanation. The TOE-Administrator can modify TSF data by formulating information flow rules and by constituting lists of information flow rules. The TSF data is safe only when consistent lists of information flow rules are constituted by the TOE-Administrator. Otherwise, due to contradicting information flow rules, the availability of user data can be jeopardised or information flows cannot be controlled as desired. The notion consistency is specified in section 2.5.1.

FMT_SMF.1 Specification of Management Functions[54]

Hierarchical to: No other components.

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions[55]:

a) *Functions that support the TOE-Administrator when preparing information flow rules.*

b) *Functions that support the TOE-Administrator when preparing consistent lists of information flow rules.*

c) *Functions that support the TOE-Administrator when preparing plausible lists of information flow rules.*

This component is refined by the following elements.

FMT_SMF.1.a The TSF shall provide the following functions that support the TOE-Administrator when preparing information flow rules:

a) Functions that enable the TOE-Administrator to use parts of already existing information flow rules when preparing new information flow rules (copy and shift).

b) Functions that allow the TOE-Administrator to assign aliases to information flow regulations and subject lists and to use them when preparing information flow rules.

c) Functions that, on the one hand, provide the TOE-Administrator with means to describe sets of data depositories in a compact way with the help of wildcards. On the other hand, the functions shall enable the TOE-Administrator to use such descriptions when preparing information flow rules.

[54] This component has been defined in the CC *Final Interpretation 065*.

[55] [assignment: list of security management functions to be provided by the TSF]

FMT_SMF.1.b The TSF shall provide the following functions that support the TOE-Administrator when preparing consistent lists of information flow rules, i.e., lists of information flow rules that meet the conditions (C1) to (C4) (see section 2.5.1):

a) Functions that enable the TOE-Administrator to get the general idea of the information flow rules adjusted up to now at any time.

b) Functions that allow the TOE-Administrator to search or sort according to different criteria in the lists of information flow rules.

c) Functions that enable the TOE-Administrator to recognise if and which certain data depositories simultaneously occur in which different information flow rules.

d) Functions that restrict the supply possibilities in an appropriate way when generating information flow rules.

FMT_SMF.1.c The TSF shall provide the following functions that support the TOE-Administrator when preparing plausible lists of information flow rules:

a) Functions that allow the TOE-Administrator to adopt predefined lists of information flow rules completely or partially.

b) Functions that enable the TOE-Administrator to review the strength (resistance) of the mechanisms that are defined in the information flow instructions.

c) Functions that enable the TOE-Administrator to check if the sequence of the individual steps determined in the information flow instructions are appropriate.

d) Functions that enable the TOE-Administrator to request or adopt a suggestion concerning the appropriate sequence of the individual steps determined in the information flow instructions.

Dependencies: No dependencies.

FMT_SMR.2 Restriction of the security roles

Hierarchical to: FMT SMR.1

FMT_SMR.2.1 The TSF shall maintain the roles *TOE-Administrator, IT-User and IT-Administrator*[56].

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

FMT_SMR.2.3 The TSF shall ensure that the conditions *(a) to (d)*[57] are met.

a. The assignment of the role TOE-administrator requires an explicit authentication.

b. The assignment of the role IT-user takes place by the following triggering events:

- *explicit authorization by the TOE-administrator.*

c. The assignment of the role IT-administrator takes place because of the following events:

i. start up of the IT system.

ii. the TOE-administrator signals the beginning of an administration activity

iii. [assignment: events that point to an administration action.]

d. The role IT-administrator and one of the roles IT-user and TOE-administrator cannot act simultaneously.

Dependencies: FIA_UID.1 Timing of identification

[56] [assignment: the authorised identified roles]
[57] [assignment: conditions for the different roles]

Application note 14. For the definition of additional events (assignment in FMT_SMR.2.3 (c)) that are signalled to the TOE by the IT environment (as, for instance, the alarm of an intrusion detection system), the ST author has to check whether a refinement of the component FPT_ITT.1 is necessary to guarantee that these signals are securely transferred to the TOE.

5.1.1.5 Class FTA: TOE access

FTA_SSL.3 TSF-initiated termination

Hierarchical to: No other components.

FTA_SSL.3.1 The TSF shall terminate an interactive session *of the TOE-Administrator*[58] after a [assignment: *time interval of user inactivity*].

Dependencies: No dependencies.

Explanation. The IT environment typically provides with security mechanisms (e.g. locking of a session or shut down of an IT system) that protect the sessions of an IT-User (see OE.NoCapture). The TOE has to provide with additional mechanisms to suitably protect sessions of a TOE-Administrator.

[58] [refinement]

5.1.2 TOE security assurance requirements

Table 8 gives an overview on the security assurance requirements that have to be fulfilled by the TOE. They correspond to the assurance level EAL2 of part 3 of the Common Criteria augmented with the component AVA_MSU.3.

No.	Component	Description
1.	ACM_CAP.2	Configuration items
2.	ADO_DEL.1	Delivery procedures
3.	ADO_IGS.1	Installation, generation, and start-up procedures
4.	ADV_FSP.1	Informal functional specification
5.	ADV_HLD.1	Descriptive high-level design
6.	ADV_RCR.1	Informal correspondence demonstration
7.	AGD_ADM.1	Administrator guidance
8.	AGD_USR.1	User guidance
9.	ATE_COV.1	Evidence of coverage
10.	ATE_FUN.1	Functional testing
11.	ATE_IND.2	Independent testing - sample
12.	AVA_MSU.3	Analysis and testing for insecure states
13.	AVA_SOF.1	Strength of TOE security function evaluation
14.	AVA_VLA.1	Developer vulnerability analysis

Table 8: TOE security assurance requirements

5.1.2.1 Class ACM: Configuration management

ACM_CAP.2 Configuration items

Dependencies: No dependencies.

Developer action elements:

ACM_CAP.2.1D The developer shall provide a reference for the TOE.

ACM_CAP.2.2D The developer shall use a CM system.

ACM_CAP.2.3D The developer shall provide CM documentation.

Content and presentation of evidence elements:

ACM_CAP.2.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.2.2C The TOE shall be labelled with its reference.

ACM_CAP.2.3C The CM documentation shall include a configuration list.

ACM_CAP.2.3+C The configuration list shall uniquely identify all configuration items that comprise the TOE.[59]

ACM_CAP.2.4C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.5C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.2.6C The CM system shall uniquely identify all configuration items.

Evaluator action elements:

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

[59] This element has been added according to the CC *Final Interpretation* 003.

5.1.2.2 Class ADO: Delivery and operation

ADO_DEL.1 Delivery procedures

Dependencies: No dependencies.

Developer action elements:

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

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

Content and presentation of evidence elements:

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

Evaluator action elements:

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

ADO_IGS.1 Installation, generation and start-up procedures

Dependencies: AGD_ADM.1 Administrator guidance

Developer action elements:

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

Content and presentation of evidence elements:

ADO_IGS.1.1C The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.[60]

Evaluator action elements:

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

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

5.1.2.3 Class ADV: Development

ADV_FSP.1 Informal functional specification

Dependencies: ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

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

[60] This element has been reworded according to CC *Final Interpretation 051*.

Content and presentation of evidence elements:

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

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

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

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

Evaluator action elements:

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

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

ADV_HLD.1 Descriptive high-level design

Dependencies: ADV_FSP.1 Informal functional specification
ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_HLD.1.1D The developer shall provide the high-level design of the TSF.

Content and presentation of evidence elements:

ADV_HLD.1.1C The presentation of the high-level design shall be informal.

ADV_HLD.1.2C The high-level design shall be internally consistent.

ADV_HLD.1.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.1.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.1.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.1.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.1.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

Evaluator action elements:

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

ADV_HLD.1.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_RCR.1 Informal correspondence demonstration

Dependencies: No dependencies.

Developer action elements:

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

Content and presentation of evidence elements:

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

Evaluator action elements:

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

5.1.2.4 Class AGD: Guidance documents

AGD_ADM.1 Administrator guidance

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

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

Content and presentation of evidence elements:

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

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

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

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

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

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

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

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

Evaluator action elements:

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

AGD_USR.1 User guidance

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

AGD_USR.1.1D The developer shall provide user guidance.

Content and presentation of evidence elements:

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

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

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

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

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

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

Evaluator action elements:

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

5.1.2.5 Class ATE: Testing

ATE_COV.1 Evidence of coverage

Dependencies:	ADV_FSP.1	Informal functional specification
	ATE_FUN.1	Functional testing

Developer action elements:

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

Content and presentation of evidence elements:

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

Evaluator action elements:

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

ATE_FUN.1 Functional testing

Dependencies: No dependencies.

Developer action elements:

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation of evidence elements:

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

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

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

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

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

Evaluator action elements:

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

ATE_IND.2 Independent testing – sample

Dependencies: ADV_FSP.1 Informal functional specification
AGD_ADM.1 Administrator guidance
AGD_USR.1 User guidance
ATE_FUN.1 Functional testing

Developer action elements:

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

Content and presentation of evidence elements:

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

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

Evaluator action elements:

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

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

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

5.1.2.6 Class AVA: Vulnerability assessment

AVA_MSU.3 Analysis and testing for insecure states

Dependencies:	ADO_IGS.1	Installation, generation and start-up procedures
	ADV_FSP.1	Informal functional specification
	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance

Developer action elements:

AVA_MSU.3.1D The developer shall provide guidance documentation.

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

Content and presentation of evidence elements:

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

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

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

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

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

Evaluator action elements:

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

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

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

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

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

AVA_SOF.1 Strength of TOE security function evaluation

Dependencies:	ADV_FSP.1	Informal functional specification
	ADV_HLD.1	Descriptive high- level design

Developer action elements:

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

Content and presentation of evidence elements:

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

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

Evaluator action elements:

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

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

AVA_VLA.1 Developer vulnerability analysis

Dependencies:	ADV_FSP.1	Informal functional specification
	ADV_HLD.1	Descriptive high-level design
	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance

Developer action elements:[61]

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

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

Content and presentation of evidence elements: [62]

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.

[61] The developer action elements have been replaced according to CC *Final Interpretation 051*.

[62] The content and presentation of evidence elements have been replaced according to CC *Final Interpretation 051*.

5.2 Security requirements for the IT environment

Table 9 shows an overview over the security functional requirements that have to be fulfilled by the IT environment.

No.	Component	Description
1a.	FCS_COP.1A	Cryptographic support
1b.	FCS_COP.1B	
1c.	FCS_COP.1C	
1d.	FCS_COP.1D	
1e.	FCS_COP.1E	
1f.	FCS_COP.1F	
2.	FPT_RVM.1	Non-bypassability of the TSP
3.	FPT_ITT.1	Basic internal TSF data transfer protection
4.	FPT_SEP.1	TSF domain separation
5.	FPT_STM.1	Reliable time stamps

Table 9: Security functional requirements for the IT environment.

Application note 15. The manufacturer shall not be restricted in the design of the cryptographic key management by this protection profile. Therefore, the dependencies that derive from the iterations of the component FCS_COP.1 are not resolved. It is the ST author's task to supplement suitable requirements on the cryptographic key management.

5.2.1 Class FCS: Cryptographic support

The choice of cryptographic algorithms that is based on the below iterations of the component FCS_COP.1 follows the BSI's obligatory stipulation of the SPHINX-specification [SPHINX, Chapter 11]. Also, it is required to support the cryptographic algorithm "Advanced Encryption Standard (AES)". In addition to the consideration of the demanded least key sizes, the choice of all the parameters (padding, choice of prime factors, random number generator, etc.) is to be designed such that the minimum strength level SOF-medium that is demanded in this protection profile is reached.

Application note 16. In addition to those that have been mentioned, more cryptographic algorithms can be supported. In this case, the recommendations of the BSI, in particular the "suitable crypto-algorithms" that are published periodically in the "Bundesanzeiger", are to be considered. For instance, one might add the schema RSAES-OAEP that has been recommended in the SPHINX-specification [SPHINX, Section 11.4.1.2].

Application note 17. All the cryptographic algorithms that are used by the TOE in principle have to reach the minimum strength level SOF-medium. To meet this demand it might be necessary to replace some of the cryptographic algorithms that are demanded in this protection profile by others, or to suitably adjust their parameters, respectively. This is the case whenever – due to the technical progress – one of the here demanded cryptographic algorithms does no longer reach the strength level SOF-medium. For the choice of alternatives the current recommendations of the BSI are to be considered. For a transitional amount of time it might be necessary to support cryptographic algorithms even if they do not any longer reach the strength level SOF-medium. In this case their use by the TOE has to be prevented in general. To be cleared, an explicit intervention by the TOE-Administrator should be required together with a clear statement concerning the weakness of the algorithm.

FCS_COP.1A Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1A.1 The *IT environment shall*[63] perform *encryption and decryption of UserData*[64] in accordance with a specified cryptographic algorithm: *AES*[65] and cryptographic key sizes *of at least 128 Bit*[66] that meet the following *standards: [FIPS 197]*[67].

Dependencies:	[FDP_ITC.1	Import of user data without security attributes
	or	
	FCS_CKM.1	Cryptographic key generation]
	FCS_CKM.4	Cryptographic key destruction
	FMT_MSA.2	Secure security attributes

FCS_COP.1B Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1B.1 The *IT environment shall*[68] perform *encryption and decryption of UserData*[69] in accordance with a specified cryptographic algorithm: *Triple-DES in CBC-Mode*[70] and cryptographic key sizes *of 128 Bit (effectively 112 bit)*[71], that meet the following *standards: [FIPS 46-3], [FIPS 81], [ISO/IEC 10116] [X9.52]*[72].

Dependencies:	[FDP_ITC.1	Import of user data without security attributes
	or	
	FCS_CKM.1	Cryptographic key generation]
	FCS_CKM.4	Cryptographic key destruction
	FMT_MSA.2	Secure security attributes

[63] [refinement: TSF shall]
[64] [assignment: list of cryptographic operations]
[65] [assignment: cryptographic algorithm]
[66] [assignment: cryptographic key sizes]
[67] [assignment: list of standards]
[68] [refinement: TSF shall]
[69] [assignment: list of cryptographic operations]
[70] [assignment: cryptographic algorithm]
[71] [assignment: cryptographic key sizes]
[72] [assignment: list of standards]

FCS_COP.1C Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1C.1 The *IT environment shall*[73] perform *encryption and decryption of message keys*[74] in accordance with a specified cryptographic algorithm: *RSA*[75] and cryptographic key sizes *of a minimum of 1024 Bit*[76], that meet the following *standards: [PKCS #1]*[77].

Dependencies:	[FDP_ITC.1	Import of user data without security attributes
	or	
	FCS_CKM.1	Cryptographic key generation]
	FCS_CKM.4	Cryptographic key destruction
	FMT_MSA.2	Secure security attributes

FCS_COP.1D Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1D.1 The *IT environment shall*[78] perform *the computation of hash values for UserData*[79] in accordance with a specified cryptographic algorithm*: SHA-1*[80] and cryptographic key sizes: *none*[81]*,* that have to meet the following *standards: [FIPS 180-1]*[82].

Dependencies:	[FDP_ITC.1	Import of user data without security attributes
	or	
	FCS_CKM.1	Cryptographic key generation]
	FCS_CKM.4	Cryptographic key destruction
	FMT_MSA.2	Secure security attributes

[73] [refinement: TSF shall]
[74] [assignment: list of cryptographic operations]
[75] [assignment: cryptographic algorithm]
[76] [assignment: cryptographic key sizes]
[77] [assignment: list of standards]
[78] [refinement: TSF shall]
[79] [assignment: list of cryptographic operations]
[80] [assignment: cryptographic algorithm]
[81] [assignment: cryptographic key sizes]
[82] [assignment: list of standards]

FCS_COP.1E Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1E.1 The *IT environment shall*[83] perform *the generation and verification of signatures for UserData*[84] in accordance with a specified cryptographic algorithm*: RSA*[85] and cryptographic key sizes *of at least 1024 Bit*[86], that meet the following *standards: [PKCS #1]*[87].

Dependencies:	[FDP_ITC.1	Import of user data without security attributes
	or	
	FCS_CKM.1	Cryptographic key generation]
	FCS_CKM.4	Cryptographic key destruction
	FMT_MSA.2	Secure security attributes

FCS_COP.1F Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1F.1 The *IT environment shall*[88] perform *generation and verification of electronic signatures for UserData*[89] in accordance with a specified cryptographic algorithm: *SHA-1 with RSA*[90] and cryptographic key *sizes of a minimum of 1024 Bit*[91], that meet the following *standards: [PKCS #1], [FIPS 180-1]* [92].

Dependencies:	[FDP_ITC.1	Import of user data without security attributes
	or	
	FCS_CKM.1	Cryptographic key generation]
	FCS_CKM.4	Cryptographic key destruction
	FMT_MSA.2	Secure security attributes

[83] [refinement: TSF shall]
[84] [assignment: list of cryptographic operations]
[85] [assignment: cryptographic algorithm]
[86] [assignment: cryptographic key sizes]
[87] [assignment: list of standards]
[88] [refinement: TSF shall]
[89] [assignment: list of cryptographic operations]
[90] [assignment: cryptographic algorithm]
[91] [assignment: cryptographic key sizes]
[92] [assignment: list of standards]

— This page intentionally left blank. —

5.2.2 Class FPT: TSF Protection

FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components.

FPT_ITT.1.1 The *IT environment shall*[93] protect *time stamps and information, crucial for the identification of demanded information flows (subject identity, operation and data depository)*[94] from *modification*[95] when such data is transmitted *to the TOE*[96].

Dependencies: No dependencies.

FPT_RVM.1 Non-bypassability of the TSP

Hierarchical to: No other components.

FPT_RVM.1.1 *The IT environment shall*[97] ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

Dependencies: No dependencies.

Explanation. This requirement ensures that the IT environment activates the TOE before users can demand information flows.

[93] [refinement: TSF shall]

[94] [refinement: TSF data]

[95] [selection: disclosure, modification]

[96] [refinement: between separate parts of the TOE]

[97] [refinement: TSF shall]

FPT_SEP.1 TSF domain separation

Hierarchical to: No other components.

FPT_SEP.1.1 The *IT environment shall*[98] maintain a security domain for *the TOE*[99] execution that protects *the TSF*[100] from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The *IT environment shall*[101] enforce separation between the security domains of subjects in the TSC.

Dependencies: No dependencies.

Explanation. FPT_SEP.1.1 serves the purpose to prevent potential attackers from modifying the internal state of the TOE in a way that circumvents, deactivates, falsifies, or invalidates the TSF.

FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

FPT_STM.1.1 The *IT environment shall*[102] be able to provide reliable time stamps for *the use by the TSF*[103].

Dependencies: No dependencies.

Explanation. FPT_STM.1 and FPT_ITT.1 ensure that reliable time stamps will be provided and that time stamps as well as the information for the identification of demanded information flows are protected against modification during transfer to the TOE. It has to be guaranteed that an IT-Administrator, who is in principle able to manipulate the IT system, is not able to manipulate - in accordance with the strength of function level required in this PP and the assumptions on the capabilities of a potential attacker assumed in the vulnerability assessment - time stamps as well as the information for the identification of demanded information flows during the transfer to the TOE. The ST author is responsible to include corresponding additional functional requirements (concerning the management of FPT), since the choice of adequate requirements depends on the specifics of the IT environment, and therefore it is impossible to fix these requirements on the fairly abstract level of this protection profile.

[98] [refinement: TSF shall]
[99] [refinement: its own]
[100] [refinement: it]
[101] [refinement: TSF shall]
[102] [refinement: TSF shall]
[103] [refinement: its own use]

6 PP application notes

The section of the protection profile contain adequately identified application notes. In addition the following is pointed out.

The TOE makes use of cryptographic functions that are implemented in hardware, firmware, and/or software to cover its security objectives. It is not demanded from the TOE itself to offer cryptographic support. The protection profile is designed in a way such that this functionality can be supplied by another trustworthy IT product (e.g. smartcards).

The manufacturer has to make sure that data to be encrypted respectively signed is suitably protected during the transmission to the components for the cryptographic support. For this purpose and where appropriate, the ST author can incorporate the following functional requirement components in the ST:

- FDP_UCT.1 (Basic data exchange confidentiality)
- FDP_UIT.1 (Data exchange integrity)
- FTP_ITC.1 (Inter-TSF trusted channel)

If the functions for the cryptographic support are entirely or partly components of the product (composite TOE) the mentioned requirement components are not, respectively only partly, needed. Because of the manifold implementation possibilities, their usage is not stipulated in this protection profile.

If the functions for the cryptographic support are entirely realised within the TOE, the component FCS_COP.1 (as well as all directly or indirectly depending requirement components) has to be shifted to the section security functional requirements on the TOE. Furthermore the security objectives OE.Disclosure and OE.Manipulation have to be deleted without substitution since they are entirely covered by the objectives O.Disclosure and O.Manipulation.

7 Rationale

7.1 Security objectives rationale

Table 10 shows that each security objective, identified in chapter 4. counteracts at least one threat respectively covers one assumption or organisational security policy.

Aspects of the TOE security environment	Security objectives
A.NoBypass	OE.NoBypass
A.Selection	OE.Selection
A.Qualification	OE.Qualification
A.NoCapture	OE.NoCapture
A.NoVirus	OE.NoVirus
T.InformationFlow	O.InformationFlow, O.Support, O.TOE-Administration, OE.Qualification, OE.Selection, OE.NoBypass
T.Read	O.InformationFlow, O.Disclosure, OE.Disclosure
T.Spy	O.InformationFlow, O.Disclosure, OE.Disclosure, O.Impersonate, OE.NoCapture, OE.NoBypass, OE.NoVirus, OE.Selection
T.Manipulate	O.InformationFlow, O.Manipulation, OE.Manipulation
T.Write	O.InformationFlow, O.Manipulation, OE.Manipulation, O.Impersonate, OE.NoCapture, OE.NoBypass, OE.NoVirus, OE.Selection
T.Unaware	O.InformationFlow, O.Disclosure, OE.Disclosure, O.Manipulation, OE.Manipulation, O.Support, O.TOE-Administration, OE.Qualification
T.Modification	O.TOE-Administration, O.Impersonate
T.Confidentiality	O.TOE-Administration, O.Impersonate, OE.Qualification
T.Impersonate	O.Impersonate, OE.NoCapture
T.Support	O.Support, OE.Qualification
P.Appropriation	O.InformationFlow, O.Support, O.TOE-Administration, OE.Qualification, OE.NoBypass, OE.Selection

Table 10: Coverage of the TOE security environment by the security objectives

In what follows it is explained for each aspect of the security environment as presented in Chapter 3 why it is covered by the security objectives listed in Table 10.

A.NoBypass

The wording clearly shows that the assumption A.NoBypass is directly covered by the security objective OE.NoBypass.

A.Selection

The wording clearly shows that the assumption A.Selection is directly covered by the security objective OE.Selection.

A.Qualification

The wording clearly shows that the assumption A.Qualification is directly covered by the security objective OE.Qualification.

A.NoCapture

The wording clearly shows that the assumption A.NoCapture is directly covered by the security objective OE.NoCapture.

Explanation. Note that A.NoCapture guarantees that a session, once started by an authorized user, cannot be taken over by another person. However, it does not avoid that another user can start a new session on behalf of that user. The start of such a new session can be avoided, for instance, if one combines the concept of the distinction between a maintenance mode and a standard mode (see Application note 4) with an explicit authorisation of the IT-user by the TOE-Administrator.

A.NoVirus

The wording clearly shows that the assumption A.NoVirus is directly covered by the security objective OE.NoVirus.

T.InformationFlow

The security objective O.InformationFlow ensures that information flows can proceed only in accordance with the specified security policy. The objectives O.Support, O.TOE-Administration and OE.Qualification make sure that the specified security policy corresponds to the IT-User’s need for protection. Finally, OE.Selection and OE.NoBypass guarantee that no information flow may take place without being noticed by the TOE.

T.Read

By encryption, the security objectives O.Disclosure respectively OE.Disclosure in co-operation with O.InformationFlow prevent from any violation of the confidentiality of UserData.

T.Spy

The security objectives OE.NoCapture, O.InformationFlow and O.Impersonate ensure that unauthorised users respectively the IT-Administrator may not directly trigger any information flows with controlled objects. In connection with OE.NoBypass and OE.Selection the objectives O.InformationFlow and O.Disclosure respectively OE.Disclosure make sure that malicious software may trigger information flows violating the confidentiality of UserData only if it acts as an integral part of a controlled subject. This is avoided by OE.NoVirus.

T.Manipulate

Using electronic signatures, the security objectives O.Manipulation respectively OE.Manipulation in cooperation with O.InformationFlow prevent any unnoticed violation of the integrity or authenticity of UserData during a proceeding information flow.

T.Write

The security objectives OE.NoCapture, O.InformationFlow and O.Impersonate ensure that unauthorised users respectively the IT-Administrator may not directly trigger any information flow with controlled objects. In connection with OE.NoBypass and OE.Selection the objectives O.InformationFlow and O.Manipulation respectively OE.Manipulation make sure that malicious software may trigger information flows unnoticed and thus violating the confidentiality of UserData only if it acts as an integral part of a controlled subject. This is avoided by OE.NoVirus.

T.Unaware

In connection with O.InformationFlow, the security objectives O.Disclosure respectively OE.Disclosure and O.Manipulation respectively OE.Manipulation ensure the realisation of the specified security policy in order to protect integrity, authenticity and confidentiality of the UserData without burdening the IT-User with the enforcement of the necessary precautions. The objectives O.Support, O.TOE-Administration and OE.Qualification make sure that the specified security policy corresponds with the IT-User's need for protection.

T.Modification

The threat T.Modification has to be counteracted by protecting the integrity of the TSF data. This is ensured by the security objectives O.TOE-Administration and O.Impersonate.

T.Confidentiality

The security objectives O.TOE-Administration and O.Impersonate ensure that the TOE limits the access to the ProtocolData to the role of the TOE-Administrator. In cooperation with the security objective OE.Qualification this ensures that other persons have no access to the ProtocolData.

T.Impersonate

The correct identification and authentication of the TOE-Administrator as a base for the assignment of roles in the TOE as well as the preservation of the correctness of this role assignment is ensured by O.Impersonate. The security objective O.Impersonate makes sure that the role IT-User is assigned correctly and, in cooperation with OE.NoCapture, that the correctness of this role assignment is preserved.

T.Support

The threat T.Support has to be counteracted by supporting the TOE-Administrator with the administration of the TOE. This is ensured by the security objectives O.Support and OE.Qualification.

P.Appropriation

The security objective O.InformationFlow ensures that UserData may be processed only by the subjects mentioned for this purpose in the security policy. The objectives O.Support, O.TOE-Administration and OE.Qualification make sure that the specified security policy corresponds with the IT-User's need for protection. Finally, OE.Selection and OE.NoBypass guarantee that no information flow may occur unnoticed by the TOE.

7.2 Security requirements rationale

7.2.1 Security functional requirements rationale

Table 11 shows that each security functional requirement identified in Chapter 5 serves the enforcement of at least one (IT) security objective.

(IT) security objectives	Functional TOE security requirements	
	principal	supporting
O.InformationFlow	FDP_IFC.1, FDP_IFF.1, FDP_ETC.1, FDP_ITC.1	FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1A, FMT_MTD.3, FMT_SMR.2, FIA_UID.2, FPT_ITT.1, FPT_RVM.1, FPT_SEP.1, AVA_MSU.3
O.Disclosure	FDP_IFC.1, FDP_IFF.1, FDP_ETC.1, FDP_ITC.1	FMT_MTD.1A, FMT_MTD.3, FCS_COP.1A – FCS_COP.1C, AVA_MSU.3
O.Manipulation	FDP_IFC.1, FDP_IFF.1, FDP_ETC.1, FDP_ITC.1	FMT_MTD.1A, FMT_MTD.3, FCS_COP.1D – FCS_COP.1F, AVA_MSU.3
O.Support	FDP_IFC.1, FDP_IFF.1, FAU_GEN.1, FAU_SAR.1, FAU_SAR.3, FAU_SEL.1, FMT_MSA.3, FMT_MTD.3, FMT_SMF.1	FAU_STG.1, FAU_STG.3, FMT_MSA.1, FMT_MTD.1A, FMT_MTD.1B, FPT_STM.1, FPT_ITT.1, AVA_MSU.3
O.TOE-Administration	FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FIA_UAU.1, FIA_UID.1, FTA_SSL.3, FMT_MOF.1, FMT_MSA.1, FMT_SMR.2, FMT_MTD.1A, FMT_MTD.1B	FPT_RVM.1, FPT_SEP.1
O.Impersonate	FIA_UAU.1, FIA_UID.1, FTA_SSL.3, FIA_UID.2, FMT_SMR.2	
OE.Disclosure	FCS_COP.1A – FCS_COP.1C	
OE.Manipulation	FCS_COP.1D – FCS_COP.1F	
OE.NoBypass	FPT_RVM.1, FPT_SEP.1	
OE.Selection	FPT_ITT.1, FPT_STM.1	

Table 11: Coverage of the (IT) security objectives by security requirements

Consecutively, it is explained for each of the (IT) security objectives identified in Chapter 4 why it is met by the security functional requirements listed in Table 11. For the security objective OE.Qualification there is no such explanation, as it is a security objective for the environment not referring to IT. No security functional requirements are assigned to the security objectives OE.NoCapture and OE.NoVirus, since such requirements do not contribute to the enforcement of the security objectives of the TOE.

Application note 18. The manufacturer has to supply evidence that the security objectives OE.Qualification, OE.NoCapture and OE.NoVirus are fulfilled. They can be achieved by suitable training measures and by implementing IT baseline protection measures.

O.InformationFlow

The components FDP_IFC.1, FDP_IFF.1, FDP_ETC.1 and FDP_ITC.1 ensure that the demanded information flows within the IT system as well as out of the IT system respectively into it are controlled in accordance with the security policy specified by the RuleData. The components FMT_MTD.1A and FMT_MTD.3 particularly assure the reliability of the RuleData regarding the compatibility of the specified information flow regulations.

The components FMT_MTD.1A, FMT_MTD.3, FMT_MSA.1, and FMT_MSA.3 assure the reliability of the security attributes. The components FPT_ITT.1, FIA_UID.2, and FMT_SMR.2 assure the reliability of information and role assignment with which, and according to FDP_IFF.1, it is decided whether a demanded information flow shall be permitted or rejected.

The component FMT_MOF.1 ensures, supported by FPT_RVM.1 and FPT_SEP.1, that the functions to enforce the TOE security policy are always active and cannot be circumvented.

The component AVA_MSU.3 contributes to the enforcement of the TOE security policy by supporting the IT-User with the necessary interaction (information concerning rejected information flows and resulting measures).

O.Disclosure

The components FDP_IFC.1, FDP_IFF.1, FDP_ETC.1 and FDP_ITC.1 ensure that demanded information flows within the IT system as well as out of the IT system respectively into it are controlled in accordance with the security policy specified by the RuleData. The components FMT_MTD.1A and FMT_MTD.3 ensure in particular the reliability of the RuleData regarding the compatibility of the specified information flow instructions.

The encryption methods which are provided by the IT environment by means of the components FCS_COP.1A – FCS_COP.1C guarantee that the information flow instructions that are to be considered according to FDP_IFF.1 and that point to the protection of the confidentiality of UserData are correctly implemented.

The component AVA_MSU.3 contributes to the enforcement of the TOE security policy by supporting IT-Users during necessary interactions (information concerning failing encryption/decryption and resulting measures).

Requirement components for the reliable transfer of UserData to the cryptographic modules, as e.g. FDP_UCT.1, FDP_UIT.1, and FTP_ITC.1, are no constituents of this protection profile. These components are needed if the cryptographic support does not belong to the TOE (Component TOE). Their consideration can be abandoned in this protection profile, however, because they contribute merely by supporting the enforcement of the security objective O.Disclosure and because they are not needed for the integration of the cryptographic support into the TOE (see Chapter PP-Application Notes).

O.Manipulation

The components FDP_IFC.1, FDP_IFF.1, FDP_ETC.1 and FDP_ITC.1 ensure that demanded information flows within the IT system as well as out of the IT system respectively into it are controlled in accordance with the security policy specified by the RuleData. The components FMT_MTD.1A and FMT_MTD.3 particularly ensure the reliability of the RuleData regarding the compatibility of the specified information flow instructions.

The signature methods which are provided by the IT environment by means of the components FCS_COP.1D – FCS_COP.1F guarantee that the information flow instructions that are to be considered according to FDP_IFF.1 and that point to the protection of the integrity and the authenticity of UserData are correctly implemented.

The component AVA_MSU.3 contributes to the enforcement of the TOE security policy by supporting IT-Users during necessary interactions (information concerning failed generation/testing of electronic signatures/certificates and resulting measures).

Requirement components for the reliable transfer of UserData to the cryptographic modules, as e.g. FDP_UCT.1, FDP_UIT.1, and FTP_ITC.1, are no constituents of this protection profile. These components are needed if the cryptographic support does not belong to the TOE (Component TOE). Their consideration can be abandoned in this protection profile, however, because they contribute merely by supporting the enforcement of the security objective O.Manipulation and because they are not needed for the integration of the cryptographic support into the TOE (see chapter PP application notes).

O.Support

The components FDP_IFC.1, FDP_IFF.1, FAU_GEN.1, FAU_SEL.1 make sure that decisions on demanded information flows are recorded in accordance with the provision taken in the RuleData.

FAU_SAR.1 and FAU_SAR.3 guarantee that the TOE-Administrator is able to analyse the ProtocolData appropriately, in order to validate the RuleData accordingly. Component FPT_STM.1 provides the temporal informations, needed for the analysis. Their protected transfer to the TOE is ensured by the component FPT_ITT.1. FAU_STG.1, FAU.STG.3 and FMT_MTD.1B guarantee the integrity of the ProtocolData underlying such an analysis and validation.

FMT_MSA.3, FMT_MTD.3 and FMT_SMF.1 vitally contributes to simplifying the TOE-Administrator's administration of the RuleData, e.g., by preventing the usage of inconsistent lists of information flow rules and by allowing the reuse of well-tried solutions. Component AVA_MSU.3 guarantees the quality of the management functions demanded by FMT_SMF.

FMT_MSA.1 and FMT_MTD.1A support the TOE-Administrator by helping to ensure that the RuleData and security attributes which are crucial for decisions concerning policy may not be altered without his/her knowledge.

O.TOE-Administration

The interplay of the components FIA_UID.1, FIA_UAU.1, FTA_SSL.3, FMT_SMR.2, FAU_SAR.1, FAU_SAR.2, and FMT_MTD.1B, supported by FPT_SEP.1, guarantees that only an user acting in the role TOE-Administrator is able to take note of the protocol data. Due to FAU_SAR.1 and FAU_SAR.3 the TOE-Administrator can analyse the protocol data appropriately.

The components FIA_UID.1, FIA_UAU.1, FTA_SSL.3, FMT_SMR.2, and FMT_MOF.1 ensure that only the TOE-Administrator is able to deactivate the TOE. This is supported by the components FPT_SEP.1 and FPT_RVM.1 which are provided by the environment.

The components FIA_UID.1, FIA_UAU.1, FTA_SSL.3, FMT_SMR.2, FMT_MSA.1, and FMT_MTD.1A, supported by FPT_SEP.1, ensure that the RuleData and security attributes that are important for policy decisions can only be changed by the role TOE-Administrator.

O.Impersonate

The components FIA_UID.1, FIA_UAU.1, FTA_SSL.3, and FMT_SMR.2 ensure that only an authorised person can act in the role TOE-Administrator. The correctness of the role assignment IT-User and IT-Administrator is guaranteed by the components FIA_UID.2 and FMT_SMR.2. The component FMT_SMR.2 ensures that the role IT-Administrator and one of the roles IT-User and TOE-Administrator cannot act simultaneously.

OE.Disclosure

The IT environment provides with appropriate cryptographic functions (especially: encryption procedure) according to FCS_COP.1A – FCS_COP.1C. They allow for the encryption of UserData in a way that they are protected from unauthorised notice during the transfer.

OE.Manipulation

According to FCS_COP.1D – FCS_COP.1F the IT environment provides appropriate hash-procedures, procedures for the generation of electronic signatures and procedures to validate electronic signatures and certificates. An appropriate application of hash-procedures and procedures for the generation of electronic signatures make it possible to protect user data within the IT system or during transfer from unnoticed modification and to attach authenticity proofs. The procedures for validating electronic signatures and certificates can be applied to verify the integrity and authenticity of UserData.

OE.NoByPass

The component FPT_RVM.1 ensures that the TOE is always active. The component FPT_SEP.1 allows the protection of the TSF data in a way that the TOE operates as intended. Therefore it is guaranteed that all information flows are controlled by the TOE.

OE.Selection

The component FPT_STM.1 ensures that time stamps provided by the IT environment are reliable. The component FPT_ITT.1 guarantees that these and all other information needed to identify demanded information flows is protected against modification during transfer to the TOE. This way it is ensured that the provided information is correct.

7.2.2 Dependencies of security functional requirements

Table 12 gives an overview on the security functional requirements of this protection profile together with their dependencies. For each dependency it is specified whether and by which other functional requirements of this protection profile they can be resolved. It should be noted that the dependencies of FCS_COP.1A – FCS_COP.1F to FDP_ITC.1 and FCS_CKM.1 are to be fulfilled alternatively. For all other alternative dependencies only the chosen alternative is given.

The dependencies are resolved for all components except FMT_MTD.3 and FCS_COP.1A – FCS_COP.1F.

FMT_MTD.3

The dependency of ADV_SPM.1 has not been resolved because a clear definition of the secure values is given by the refined element FMT_MTD.3.a. The reason for this definition arises directly from the specification of the TOE security policy (see Chapter 2.5). According to the Common Criteria[104] this dependency can be argued away.

FCS_COP.1A – FCS_COP.1F

The dependencies of FCS_COP.1A – FCS_COP.1F have not been resolved because there are manifold possible alternatives to realise the cryptographic support. In this protection profile it is thus not advisable to formulate detailed requirements on the combinations of the cryptographic support and the specified security policy. The open dependencies of FCS_COP.1 thus commit the ST author to add such requirements. This applies in particular for the important field concerning the management of cryptographic keys (family FMT_CKM).

[104] Citation from the Common Criteria, Part 2, Annex H.3, paragraph 1046: “If the developer provided a clear definition of the secure values and the reason why they should be considered secure, the dependency from FMT_MSA.2 to ADV_SPM.1 can be argued away.”

No.	CC-Component	Dependency	Resolved by
1.	FAU_GEN.1	FPT_STM.1	No. 27
2.	FAU_SAR.1	FAU_GEN.1	No. 1
3.	FAU_SAR.2	FAU_SAR.1	No. 2
4.	FAU_SAR.3	FAU_SAR.1	No. 2
5.	FAU_SEL.1	FAU_GEN.1 FMT_MTD.1	No. 1 No. 18b
6.	FAU_STG.1	FAU_GEN.1	No. 1
7.	FAU_STG.3	FAU_STG.1	No. 6
8.	FDP_ETC.1	FDP_IFC.1	No. 9
9.	FDP_IFC.1	FDP_IFF.1	No. 10
10.	FDP_IFF.1	FDP_IFC.1 FMT_MSA.3	No. 9 No. 17
11.	FDP_ITC.1	FDP_IFC.1 FMT_MSA.3	No. 9 No. 17
12.	FIA_UAU.1	FIA_UID.1	No. 13
13.	FIA_UID.1	keine	—
14.	FIA_UID.2	keine	—
15.	FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	No. 20 No. 21
16.	FMT_MSA.1	FDP_IFC.1 FMT_SMR.1 FMT_SMF.1	No. 9 No. 21 No. 20
17.	FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	No. 16 No. 21
18a.	FMT_MTD.1A	FMT_SMF.1	No. 20
18b.	FMT_MTD.1B	FMT_SMR.1	No. 21
19.	FMT_MTD.3	ADV_SPM.1	not resolved
		FMT_MTD.1	No. 18a
20.	FMT_SMF.1	keine	—
21.	FMT_SMR.2	FIA_UID.1	No. 13 (TOE-Administrator) No. 14 (IT-User, IT-Administrator)
22.	FTA_SSL.3	keine	—
23.	FCS_COP.1A – FCS_COP.1F	FDP_ITC.1 FCS_CKM.1 FCS_CKM.4 FMT_MSA.2	not resolved not resolved not resolved not resolved
24.	FPT_RVM.1	keine	—
25.	FPT_ITT.1	keine	—
26.	FPT_SEP.1	keine	—
27.	FPT_STM.1	keine	—

Table 12: Dependencies between the security functional requirements

7.2.3 Mutual support of security functional requirements

Section 7.2.1 explains the protection of the security functional requirements that are classified as "principal" by supporting security functional requirements. In particular, the enforcement of the discretionary information flow control SFP is supported

- by restricting the deactivation of the TOE to the TOE-Administrator as required from the component FMT_MOF.1.
- by the protection against circumvention of the active TOE that is to be guaranteed by the IT environment as required from the component FPT_RVM.1.
- by the protection against manipulation of the TOE that is to be guaranteed by the IT environment as required from the component FPT_SEP.1.

The component FMT_MOF.1 supports the components FDP_IFC.1 and FDP_IFF.1. The components FPT_RVM.1 and FPT_SEP.1 contribute to the support for the components FDP_IFC.1, FDP_IFF.1, FIA_UAU.1, FIA_UID.1 FTA_SSL.3 and FMT_MOF.1. Moreover, FPT_SEP.1 supports the components FAU_SAR.1, FAU_SAR.2, FMT_MSA.1, FMT_MTD.1A und FMT_MTD.1B.

Furthermore, Section 7.2.1 justifies the omission of supporting CC components for the formulation of requirements on trusted channels to exchange data with external cryptographic modules.

In Section 7.2.2 the dependencies of the functional components are examined and justifications are given for dependencies that have not been resolved.

Additionally, it is explained that the operations that have been carried out are coordinated.

Selection operations

All selection operations and in particular the selection of the audit degree „minimal" (FAU_GEN.1) and the selection of default values for security attributes with „permissive" properties (FMT_MSA.3) are coordinated and correspond with the supposed low threatening potential.

Assignment operations

All assignment operations and in particular the regulation of the security policy which has to be enforced (FDP_ETC.1, FDP_IFC.1, FDP_IFF.1, FDP_ITC.1, FIA_UAU.1, FIA_UID.1, FMT_MSA.1, FMT_MSA.3, and FMT_SMF.1) are coordinated both mutually and with the conditions for the role assignment (FIA_UAU.1, FIA_UID.1, FIA_UID.2 and FMT_SMR.2) and specify comprehensively a consistent security service.

Iteration operations

The iteration of the component FMT_MTD.1 is necessary for the distinction between RuleData and ProtocolData. The iterated components are consistently used to resolve the dependencies (see Chapter 7.2.2).

The simultaneous utilisation of the hierarchical components FIA_UID.1 and FIA_UID.2 in the TOE is necessary since they are used to assign different roles with different authorities.

The components of the family FIA_UID are used consistently to resolve the dependencies of FMT_SMR.2 (see chapter 7.2.2 and chapter 5.1.1.3).

The iteration of the component FCS_COP.1 is necessary to distinguish the various cryptographic algorithms.

Refinement operations

All refinement operations are coordinated with

- the utilisation of the components for the IT environment (see Chapter 5.2) and
- the complex management requirements for the security policy that has to be enforced (see Classes FIA and FMT).

7.2.4 Security assurance requirements rationale

The assurance requirements according to the chosen evaluation level EAL2 are appropriate for the TOE because it is assumed that the security features shall only protect against obvious penetration attacks.

The augmentation with the component AVA_MSU.3 allows for an evaluation of the special requirements on the administration of the TOE (compare the refinement of the component FMT_SMF.1).

By choosing the specified evaluation level EAL2 the resolution of the dependencies of the assurance requirements is automatically given. No additional dependencies are requested for the component AVA_MSU.3.

7.2.5 Minimum strength of function level rationale

According to the current state of the art mechanisms are available in the field of cryptography and authentication that reach the level of strength SOF-medium. Although the security features of the TOE shall only protect from obvious penetration attacks, it has to be considered that encrypted and/or signed data is kept for rather long periods of time. The postulate SOF-medium for the minimum strength level of the functions is appropriate for the TOE when the permanent maintenance of information protection is considered.

A Glossary

Application An application program to which processes of the operating system level can be assigned. Example for an active functional unit as a *subject.*

Consistent list of information flow rules List of *information flow rules* with special properties that guarantee among others that only one *most specific information flow rule* exists for every *data depository* and that *information flow instructions* do not contradict each other.

Controlled data depository Data depository for which a *most specific information flow rule* exists. In this rule the control flag is set.

Data depository The unambiguous description of a depository where a passive unit (an *object*) is located. The information of the location can relate to an *object* located on a local memory medium or to an *object* accessible via a net connection.

Deactivation of the TOE The TOE is deactivated, if the TOE is unable to control occurring information flows with respect to the fixed information flow rules. Since the TOE works in a transparent manner, its deactivation has to be transparent, as well. This means that, similarly as for the active TOE, a user is unable to recognize that the TOE is deactivated. The latter has the following consequence, for instance: data which has been encrypted by the TOE has to be decrypted before the deactivation of the TOE. Consequently, the deletion of the program code of the TOE is different from its deactivation, since in this case the encrypted data will not be decrypted. Moreover, prior to the deinstallation, the TOE has to be deactivated.

Flag Binary attribute which can take the values "True" or "False".

Information Data linked to *objects*.

Information flow A flow of *information* as a result of an *operation* caused by a *subject*. Considered are read or write operations of information from/in *objects*.

Information flow instruction Instruction that specifies the type and the order of operations that have to be carried out before *information* is read out of an *object* or written in an *object*.

Information flow rule Rules that set the basis for decisions made by the *TOE* whether a demanded *information flow* shall be permitted or rejected. They specify, among others, which *subjects* have the permission to write *information* in *objects* at *data depositories* that are to be controlled or read *information* from *objects* at *data depositories* that are to be controlled and which *information flow instructions* should be considered.

IT-Administrator Role that authorises the administration of the *IT system* and the installation of the *TOE*.

IT system The system in its entirety, consisting of hardware and software components, on which the *TOE* is installed and on which the *TOE security policy* shall be enforced.

IT-User Role that authorises the usage of the *IT system*.**Maintenance** All activities concerning an *IT system* that ensure the intended functioning of the *IT system.*

Most specific information flow rule *Information flow rule* R in which the concerning *data depository* is mentioned. For this *information flow rule* applies that no *information flow rule* exists in which, besides this *data depository*, only a few of the *data depositories* mentioned in R are named.

Objects Passive units that can contain *information*. They are the target of *operations* being carried out by *subjects*.

Operating system The part of an *IT system* that is responsible for the resource administration.**ProtocolData** The ProtocolData comprise all events audited by the TOE. This includes in particular permitted and rejected information flows.

Role Defines the permitted activity of a class of *TOE* users. One user ,however, can hold more than one role simultaneously (in the extreme even all).

RuleData Part of the *TSF data* comprising the list of the employed *information flow rules.*

Security attribute Attributes assigned to s*ubjects, information* and/or *objects* in order to define a *security functional policy.*

Security functional policy A subset of the *TOE security policy* that specifies the *objects/information*, *subjects* and *operations* that are to be controlled within its application area.

Selection function A function that selects the rule out of a list of *information flow rules*, by means of which it is decided whether the information flow shall be permitted or rejected.

Subjects Declarations that are necessary to describe the active units (e.g. processes assigned to *applications*) within the *TOE*.

TOE Target of evaluation – here it is a security product which can be realised as a pure software solution as well as a combination of hardware and software components.

TOE-Administrator Role which authorises the administration of the *TOE* and the reading of the *ProtocolData*.

TOE security policy The totality of the *security functional policies* defines the TOE security policy.

Trojan Horse A malicious program that pretends being inoffensive in order to harm the *IT system* without being noticed by an *IT-User*.

TSF data Information that is used for decisions in the scope of the *TOE security policy*. *ProtocolData* and *RuleData* belong among others to the TSF data.

Unauthorised user Users of the *IT system* who are not authorised to act in the role of an *IT-User*, *IT-Administrator* or *TOE-Administrator*.

UserData Information with which the users can carry out operations and which is not used for decisions within the scope of the *TOE security policy.* It is an information that is processed by IT-Users in the scope of their activity.

B Abbreviations

ANSI	American National Standards Institute
BSI	Bundesamt für Sicherheit in der Informationstechnik
CC	Common Criteria
CEM	Common Evaluation Methodology
EAL	Evaluation Assurance Level
E-	Electronic-
FIPS	Federal Information Processing Standards
IEC	International Electrotechnical Commision
ISO	International Organisation of Standards
IT	Information Technology
LAN	Local Area Network
MU	Multi-User
PKCS	Public-Key Cryptography Standards
PP	Protection Profile
SFP	Security Functional Policy
ST	Security Target
SU	Single-User
TOE	Target of Evaluation
TSF	TOE Security Functions
TSP	TOE Security Policy
WAN	Wide Area Network

C References

[DIC-MU] Bundesamt für Sicherheit in der Informationstechnik (BSI) [German Information Security Agency]. Discretionary Information Flow Control (MU). Common Criteria Protection Profile BSI-PP-0008, Version 2.01, September 4, 2002.

[FIPS 46-3] FIPS Publication 46-3. Data Encryption Standard (DES). October 25, 1999.

[FIPS 81] FIPS Publication 81. DES Modes of Operation. December 2, 1980.

[FIPS 180-1] FIPS Publication 180-1. Secure Hash Standard (SHS). April 17, 1995.

[FIPS 197] FIPS Publication 197. Advanced Encryption Standard (AES). November 26, 2001.

[ISO/IEC 10116] ISO/IEC 10116:1997. Modes of Operation for an n-bit block cipher algorithm. 1997.

[PKCS #1] RSA Laboratories. PKCS #1 v.2.0: RSA Cryptography Standard. October 1998.

[SPHINX] Bundesamt für Sicherheit in der Informationstechnik (BSI) [German Information Security Agency]. SPHINX Pilotversuch Ende-zu-Ende-Sicherheit: Technische Grundlagen – Tailoring MTTv2. Version 2.0, 15. August 2000.

[X9.52] ANSI X9.52-1998. Triple Data Encryption Algorithm Modes of Operation.