BİLİŞİM VE BİLGİ GÜVENLİĞİ İLERİ TEKNOLOJİLER ARAŞTIRMA MERKEZİ

ULUSAL ELEKTRONİK VE KRİPTOLOJİ ARAŞTIRMA ENSTİTÜSÜ

KERMEN PORTABLE V1.0

SECURITY TARGET

Document Code :
TEDL / SDIL :
Version : 1.7
Publication Date : 29.08.2013

Title:	Security Target Kermen Portable		
Author:	Dindar ÖZ	Status:	Release
Version:	1.7	Classification:	Public
FileName:	KermenPortable_ST.pdf	Date: 29.08.2013	

DOCUMENT HISTORY

SECURITY TARGET

Version	Date	Author	Changes To the Previous Version
1.0	28.09.2012	Dindar Öz	First Publication
1.1	21.11.2012	Dindar Öz	• EAL4 conformance changed to EAL4+(ALC_FLR.2) • PP Claim removed.
1.2	25.01.2012	Dindar Öz	• Document changes related to CC v3.1 updates.
1.3	08.03.2013	Dindar Öz	• Minor document changes.
1.4	26.04.2013	Dindar Öz	• Minor document changes
1.5	28.06.2013	Dindar Öz	• New SFR's added
1.6	28.08.2013	Dindar Öz	• CC Reference is corrected
1.7	29.08.2013	Dindar Öz	• CEM Reference is corrected

SECURITY TARGET

TABLE OF CONTENTS

INDEX OF TABLES

FIGURE INDEX

SECURITY TARGET

1. INTRODUCTION

1.1 ST Reference

ST Title:	Kermen Portable v1.0 Security Target
Product Name:	Kermen Portable
Product Version:	1.0
Assurance Level:	EAL4+ (ALC_FLR.2)
CC Version:	**3.1 R4**
ST Author:	Dindar Öz
ST Publication Date:	29/08/2013
ST Version:	1.7
Key Words:	File Encryption / Decryption

1.1.1 Operation Notation for Functional Requirements

The CC allows several operations to be performed on functional requirements; refinement, selection, assignment, and iteration. Each of these operations is used in this ST.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by the word “refinement” in bold text and the added/changed words are in bold text. In cases where words from a CC requirement were deleted, a separate attachment indicates the words that were removed. Refinements are written in italic font in this ST.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections are in bold font and inside square brackets and marked with “selection”.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments are in bold font and inside square brackets and marked with “assignment”.

The **iteration** operation is used when a component is repeated with varying operations. Iterated functional requirement components are shown with a “**/IDENTIFIER**” for the components which used more than once with varying operations.

1.2 TOE Reference

TOE Identification: Kermen Portable Version 1.0

1.3 TOE Overview

1.3.1 TOE Definition

Kermen Portable is an application for file encryption on both Microsoft Windows and Pardus platforms. The underlying platform for the evaluation is limited to the MS Windows XP and MS Windows 7 operating systems. A FIPS approved open source cryptographic library cryptoPP is used to implement all cryptographic related functions.

Kermen Portable is a file encryption application intended to protect cofidential information when transmitting and storing them in unprotected environments.

1.3.2 TOE usage and security features for operational use

Kermen Portable is used for secure transmisson of files over insecure channels. This files can be any kind of confidential material. Kermen Portable encrypts those files with the users' certificates based on RFC 5652 standard. The encrypted files can only be decrypted by the users' corresponding private keys which are only accessible by the users themselves. Kermen Portable also signs the encrypted files with users' signing certificates which protects the integrity of the files. Thus by using Kermen Portable, sensitive and confidential data are protected against unauthorized reads and modifications.

1.3.3 Non-TOE hardware/software/firmware required by the TOE

Kermen Portable requires Windows XP or newer versions of Microsoft windows installed on the target machine. The target machine is assumed to be in a secure environment which is not accessible by unauthorized people. The minimum system requirements for the target machine ;

- Intel Celeron/Pentium II coequal processor
- 512 MB memory
- 500 MB harddisk space

1.4 TOE Description

This section describes the Target of Evaluation (TOE) in terms of the class of product, the operational environment, and the provided security functionality. This chapter provides a general description of the product without focusing on the evaluated configuration.

1.4.1 Introduction

Kermen Portable is a software only product for file encryption in both Microsoft Windows and Pardus environments, running on a single user PC. The program requires X.509 certificates in order to perform encryption operations.

Kermen Portable imlements asymmetric encryption by using EnvelopedData format defined in [RFC 5652] to ensure the confidentiality of the information. It also signs the confidential data before encrypting in order to detect any loss of integrity. The signing operation is performed again in accordance with the CMS standard defined in the [RFC 5652]. Encrypted and signed files can be decrypted and their signatures are validated by Kermen Portable according to the CMS standard.

X.509 certificates are used to identify individuals and perform asymmetric encryption. Kermen Portable checks the validity of the certificates upon encryption and decryption operations. In order a certificate to be accepted as valid by Kermen Portable,besides the other structural controls, the certificate chain must end up with a trusted certificate defined in the certificate store. The trusted certificates in the store are signed with a private key which is securely stored in a smartcard belonging to a trusted authority and the public key to validate the signature is embedded in the source code of the application which is not accessible by the user. Therefore, only the trusted authority can define which certificates will be accepted as trusted certificates. Certificate validation is performed according to the steps described in [RFC 5280].

Kermen Portable stores also the X.509 user certificates locally in its own certificate store.The private keys of asymmetric encryption and signing can be either in smartcards or in the certificate store as encrypted by user defined passwords.User is responsible for the confidentiality of the stored password. In case of smartcards, the security of the private keys is provided by the smartcards.

Kermen Portable provides a viewer for the certificate store. By using this viewer, users can view the stored certificates , import and export them if they need.

Kermen Portable does not remove the confidential data after encrypting it. The secure removal of the confidential data is out of the scope of this TOE.

1.4.2 TOE Definition Scope

The target of evaluation is limited to the software application Kermen Portable, version 1.0, developed by TÜBİTAK BİLGEM. Kermen Portable consist of three parts: Kermen SUR which is a file encryption GUI, Kermen DEPO which is the viewer GUI for the certificate store and the underlying API responsible for all of the cryptographic work. API depends on Crypto++ and OpenSLL crypto libraries which are open source.

The GUI applications do only forward user request to the API and they can not implement any cryptographic mechanism or other operations. For example, Kermen SUR's the only job is to identify for the underlying API what operation to perform on which file. Therefore the GUI components is not security critical. In the following picture the architecture of TOE Kermen Portable and its boundaries are shown.

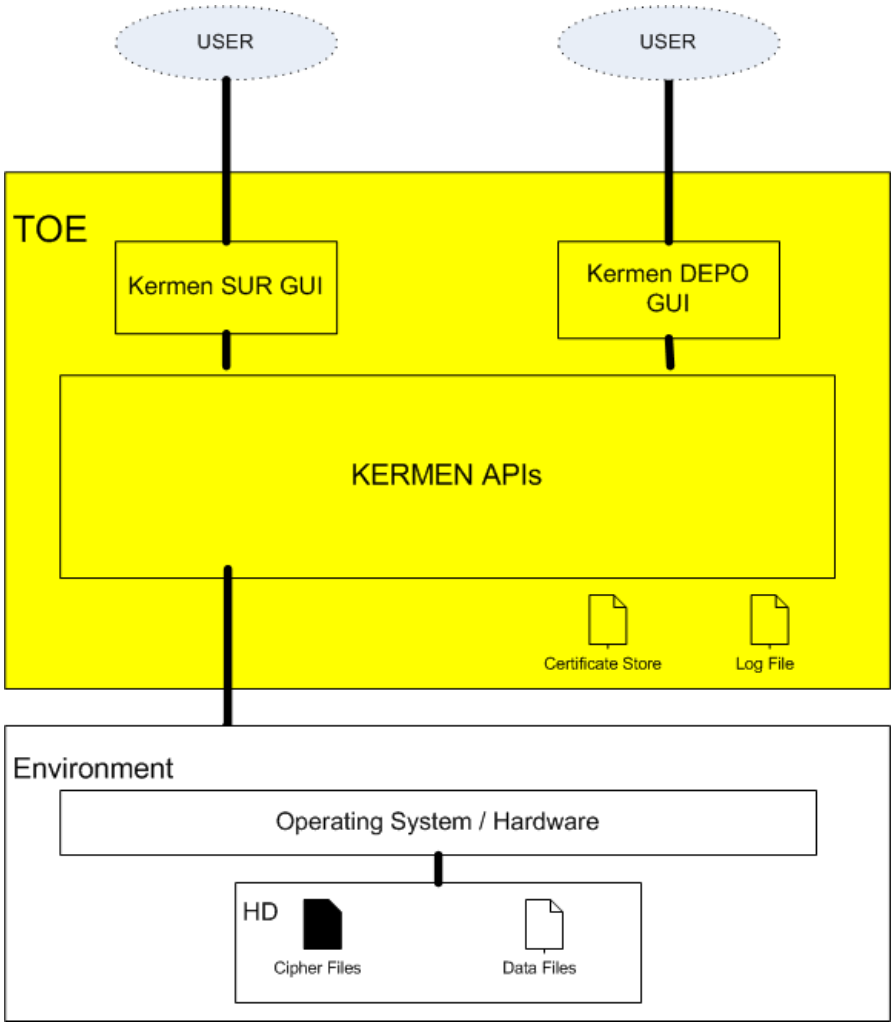

Figure 1 TOE Architecture and Boundaries

The physical scope of the TOE includes digital files of the application executables and APIs , user's encrypted files and log files. These files are assumed to be protected by the physical medium they are stored and by the underlying operating system.

1.4.3 Supported Platforms And Environment

The underlying platform for the evaluation is limited to the MS Windows XP and MS Windows 7 operating systems. No additional special equipment or infrastructure is needed.

1.4.4 Installation

No installation required in order to run Kermen Portable. Only copying program files is sufficient.

1.4.5 Configuration

Most security configurations of the application can not be configured at run time. The algorithms used for encryption and key sizes all determined in the development process and constantly defined in the application code. The password for the certficate store is specified by the user by using Kermen DEPO.

User can also set up where the encrypted files will be copied , for whom the files will be encrypted , by whom the encrypted files will be signed , and other file operation options by using Kermen SUR.

1.4.6 TOE Operation and Use

1.4.6.1 Intended Use

By using Kermen Portable, two or more individuals can exchange electronic documents securely over unprotected communication paths, eg. Networks, without risking any unauthorized persons reading the document. To achieve this, the documents are encrypted before they are sent between the two parties, and are thus made unreadable for anyone who are not included in the recipients of the document defined by the sender.
The file encryption interface of Kermen Portable is Kermen SUR application. With Kermen SUR, after configuration, encryption of files performed by the user in just one step. Dragging the files that will be encrypted onto the application window is sufficient for encyption operation to be performed.
Likewise, in order to decrypt files that are encrypted for the user, all the user needs is to drag and drop the encrypted files onto the application window. If the user's private key stored in a smartcard then that card must be plugged in for decryption and also while encrypting, to sign the document with the user's private signing key, his or her smartcard is needed. For the keys stored in the certificate store, store password specified by the user is prompted to the user.

1.4.6.2 Security Roles

The TOE is not aware of any user roles or even the concept of users, so any user with access to the TOE or the TOE environment is able to perform any operation.

1.4.6.3 Security Functionality

The following security features are provided by the TOE:

- **Encryption of files:** Files are encrypted using the EnvelopedData method defined in [RFC 5652]. In EnvelopedData, data is symmetrically encrypted and the encryption key is asymmetricallay encrypted for the recipients with their asymmetric public keys. The symmetric encryption algorithm is AES-CBC with a 256 bit key and asymmetric encryption algorithm is defined by the certificates of the recipients. It can be either RSA with defined key size or Elliptic Curve.
- **Integrity protection of files:** To detect the loss of integrity of data files , encrypted files are signed by the senders private key just before the encryption. For signing files, again, cryptographic message syntax is used and SignedData is created as described in [RFC **5652].** The signature algorithm is defined by the sender's public key placed in the user's signature certificate.
- **Decryption of files:** Encrypted files are decrypted if the user posseses the corresponding private key in his smart card or certificate store. Only those users who are defined as the recipient during the encryption process, can decrypt files due to the security functions of asymmetric encryption.
- **Integrity Check of files:** Since the encrypted files are also signed by the encryptor, After decrpytion of the file the result is a signed file to be validated. By this validation the inegrity of the original data file is ensured.
- **Nonrepudiation of files:** The private keys used in signing files are associated with a X.509 certificate therefore the identity of the user. By the nature of the signature operation,

user can claim neither that the data is not signed by himself nor that it is modified after signing.

- **Key Storage:** Soft private keys are stored in the certificate store in password based encrypted form. For this purpose , the algorithm PBKDF2 desribed in [RFC 2898] (PKCS#5) is used.
- **Import/Export Keys:** Certificates and associated private keys can be imported into and exported from the certificate store by using Kermen DEPO GUI. In both cases, the user must provide the store password.

1.4.7 TOE Environment and Physical Protection

The TOE is expected to be operated as a single user machine in a physically secure and well managed environment without a direct connection to an untrusted network.

1.5 Related Standards and Documents

[CC]	Common Criteria (CC) for Information Technology Security Evaluation, Version 3.1, revision 4, September 2012. » Part 1: Introduction and general model. September 2012. Version 3.1. CCMB-2012-09-001 » Part 2: Functional security components. September 2012. Version 3.1. CCMB-2012-09-002 » Part 3: Assurance security components. September 2012. Version 3.1. CCMB-2012-09-003
[CEM]	Common Methodology for Information Technology Security Evaluation. Evaluation Methodology. September 2012. Version 3.1, Revision 4, CCMB-2012-09-004
[GPPS]	ISO/IEC TR 15446:2004(E), Guide for the production of Protection Profiles and Security Targets, First edition 2004-07-01
[RFC 5280]	Request For Comments: 5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008
[RFC 5652]	Request For Comments: 5852, Cryptographic Message Syntax (CMS), September 2009
[RFC 2898]	Request For Comments: 2898, PKCS #5: Password-Based Cryptography Specification Version 2.0 September 2000
[RFC 4056]	Request For Comments: 4056, Use of the RSASSA-PSS Signature Algorithm in Cryptographic Message Syntax (CMS), June 2005
FIPS 197	AES Encryption Standart
FIPS 140	Security Requirements for Cryptographic Modules
FIPS 140-2 Annex.A	Approved Security Functions for FIPS PUB 140-2

2. CONFORMANCE CLAIM

2.1 CC Conformance Claim

This ST claims conformance to

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2012-09-001, Version 3.1, Revision 4, September 2012, [1]
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012, [2], Comformant
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; CCMB-2012-09-003, Version 3.1, Revision 4, September 2012, [3], Comformant

The

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; CCMB-2012-09-004, Version 3.1, Revision 4, September 2012, [4]

has to be taken into account.

2.2 PP Claim

This ST does not claim any conformance to any protection profile.

2.3 Package Claim

EAL4+ (ALC_FLR.2)

2.4 Conformance Rationale

An assurance requirement of EAL4+ (ALC_FLR.2) is required for this type of TOE since it is intended to defend against attackers with attack potential of enhanced basic. In order to provide a meaningful level of assurance that the TOE provides an adequate level of defense against such attacks, the evaluators should have access to the low level design and source code. The lowest for which such access is required is EAL4+ (ALC_FLR.2). The augmentation (ALC_FLR.2) is selected as it is easily achievable since it requires no specialist techniques on the part of the developer.

3. SECURITY PROBLEM DEFINITION ENVIRONMENT

3.1 Secure Usage Assumptions

The following conditions are assumed to exist in the TOE operational environment.These assumptions include essential environmental constraints on the secure use of the TOE. Assumptions about the intended usage of the TOE are not made.

A.SINGLE The TOE runs on a single user machine with access protected by the TOE environment; i.e., only authorised users of the TOE environment may access the TOE. This includes access control provided by the operating system or equivalent and protection against malware.

A.KEYDIS It is assumed that private keys used for decrypting and signing files are of high quality and are not disclosed to unauthorized users.

A.PASSWORDDIS It is assumed that passwords used for accessing the private keys in the certificate store and in the smart card are of high quality and are not disclosed to unauthorized users.

A.PHYSICAL The TOE is operated in a physically secure and well managed environment.

A.USER The TOE user is trustworthy and trained to manage and perform encryption of classified information in accordance with any existing security policies and information classification policies. This means especially that he knows how to classify information and how to deal with, e.g.,encrypting all files containing sensitive information with the appropriate key before exporting the file out of the TOE and/or its TOE environment.

A.CONNECT The single user PC on which the TOE is running is not connected directly to an untrusted network. This means that the PC is either assumed not to be connected to any networks or it is connected to a trusted network which is protected against attacks, so that no undocumented security critical side effects on the security functions of the TOE, which are resided in the PC, are assumed coming from this network.

3.2 Organizational Security Policies

P.ALGORITHM The TOE shall only allow the use of approved encryption algorithms and key lengths.

3.3 Threats

The threats described in this chapter are addressed by the TOE.

3.3.1 Assets and Agents

The assets and user agents used for the definition of threats are defined in the following tables.

Asset	Description	Type of Data
Data Files (primary asset)	Any file that contain the information to be protected.	User Data
Smartcard Password	The password for the smarcard login	User Data
Certificate Store Password	The password for the certificate store	User Data
Private Keys	Asymmetric private keys used for file decryption and signing	User Data
Encryption Keys	Symmetric encryption keys	User Data

Table 1: Assets

Agent	Description
Attacker	An attacker who has access to any communication channel over which the integrity protected and encrypted files are transferred, e.g., networks or other paths of transmission where communication media like CDs, DVDs including the encrypted files could be shared.

Table 2: Agents

3.3.2 Threats addressed by the TOE and the TOE environment

The threats below must be countered by the TOE and the TOE environment.

Threat: T.DISCLOSE – Loss of confidentiality

Attack	An attacker of one of the communication paths over which the Kermen Portable files are transferred succeeds in accessing the content of the file, i.e. the attacker violates the confidentiality of the information included in the file. The attack is achieved by passive attacks recording encrypted data during the transfer (e.g. eavesdropping of network communication, interception of dispatch services) and decoding the encrypted data. In general the attacker has no access to the right key and has to perform cryptanalysis to reveal the underlying plain text of the encrypted file.
Asset	Data Files, smart card passwords,certificate store password, private keys, encryption keys
Agent	Attacker

Threat: T.MODIFY – Loss of integrity	
Attack	An attacker of one of the communication paths over which the Kermen Portable files are transferred modifies the file, i.e. replacing or changing the content of the file in a way that is not detected. The attack is achieved by interrupting the transfer due to possess the file to accomplish an active attack violating the integrity of the information included in the file before sending it to the receiver. Therefore the attacker has either to break the integrity protection of the file, modifying the content of the file and reconstructing the protection again. Or the attacker replaces the whole file and constructs the integrity protection. Afterwards the file is sent to the intended destination. In both cases the attacker has to possess either the right key used for integrity protection or he has to perform cryptanalysis to reveal the right key. For possibilities to get the right key see T.DISCLOSE.
Asset	Data Files
Agent	Attacker

Table 3 Threats addressed by the TOE

In both threats described above the primary subject of the attacks is the information included in the data files transferred over an unprotected communication path.
The attackers specified as threat agents in both threats above are assumed to possess very limited opportunity of attacks, characterized as follows:

- **Expertise:** It is assumed that the passwords used to access private key material and the keys used for signing and encrypting files have not been leaked (A.KEYDIS, A.PASSWORDDIS) and the implementation is not flowed (A. PHYSICAL). The attackers know IP and related networking protocol basics and are trying to find vulnerabilities publicly known about cryptographic algorithms (systematic weaknesses). The attacker must be familiar with the “alternative” distribution channels over which the encrypted data will be

sent. Therefore a high level of expertise is required to successfully gain the plain text from encrypted data.

- **Resources**: The resource requirements to mount an attack of the types described above are high – a very large amount of computing power, either distributed or within one unit would be required to break the encryption in an appropriate time scale, expected to do not exhaust the range of at maximum some man days. In contrast to the attack within T.DISCLOSE the attack within T.MODIFY must be launched e.g. nearly on the fly, to ensure that the attack could not be detected. Network attack tools, especially network sniffers, available on the Internet are considered to be available, too. Further the attacker has the possibility to buy the product and perform cryptanalysis on the algorithms used or disassembling and reverse engineering the TOE. Therefore it is very easy for the attacker to get information about how the TOE operates. But attackers have no access (neither physical nor over the network – A.PHYSICAL, A.CONNECT) to the TOE where the information is encrypted or decrypted.
- **Motivation:** The TOE aims to protect sensitive information during the transfer over any communication paths. So, the attackers are assumed to be motivated by high-value assets and e.g. by the fact to "hack" sensitive information.

As described above it is very easy for an attacker to get information about how the TOE is operating in general – therefore an attacker will reveal easily that he has to combine the attacks described in T.DISCLOSE and T. MODIFY to be successful in violating the integrity and/or confidentiality of the file's content. Because the transferred file is first encrypted and afterwards the encrypted file is integrity protected.
Attacks which modify the content of the transferred file without breaking the integrity protection are as well conceivable. May by an attacker completely intercepts the communication so that the file does not reach it's destination. This attacks have the same effects as errors during communication have. Preliminary the availability of the information transferred is violated. The receiver fails e.g. in validating the integrity of the file, the file will not be decrypted. This attacks will not be regarded here deliberately, because it will be detected anyway.
If vulnerabilities were present in the TOE's encryption algorithm, cryptographic functions used for integrity protection, key generating algorithm or in there implementation, this may be exploited to decrease the level of expertise or resource required for success.

The opportunity to mount all attacks depends on the fact that the transferred Kermen Portable file is in general available for an attacker.

4. SECURITY OBJECTIVES

The security objectives provide a concise statement of the intended response to the security problem. This section describes which security needs will be addressed by the TOE and which will be addressed by the TOE environment, in the form of a statement of security objectives.

4.1 Security Objectives For The TOE

The following are the IT security objectives to be met by the TOE.

O.DISCLOSE The TOE must provide mechanisms that protect the information of a transmitted files such that its content is confidentiality-protected and only accessible for authorized users.

O.MODIFY The TOE must provide mechanisms that detect if an attacker has tampered with a transmitted file (i.e. replacing or modifying the content of the file); mechanisms must be provided to detect loss of integrity of the information in the

file.

O.ALGORITHM The TOE must only allow the use of approved encryption algorithms and key lengths, i.e. AES and 256 bit.

4.2 Security Objectives for the IT and non-IT Environment

The following are the security objectives that are to be satisfied without imposing technical requirements on the TOE. That is, they do not require the implementation of functions in the TOE hardware and/or software. These security objectives are assumed to be in place in the TOE environment. They are included as necessary to support the TOE security objectives in addressing the security problem defined in the TOE security environment.
Thus, the following environmental objectives may partly be IT specific and partly related to administrative methods and/or procedural measures.

OE.KEYDIS Keys used to sign and encrypt files must be of high quality and must not be disclosed to unauthorized users.

OE.PASSWORDDIS Passwords used to acces private keys in the certificate store or smartcards must be of high quality and must not be disclosed to unauthorized users.

OE.SINGLE The TOE must be run on a single user machine with access to the TOE protected by the TOE environment; i.e., only authorised users of the TOE environment have access to the TOE. This includes access control provided by the operating system or equivalent and protection against malware.

OE.PHYSICAL The TOE must be operated in a physically secure and well managed environment.

OE.USER The TOE User is trustworthy and trained to perform all actions in accordance with any existing security policies and information classification policies.

OE.CONNECT The single user PC on which the TOE is running must not be connected directly to an untrusted network. This means that the PC must either not be connected to any networks or it must be connected to a trusted network, which is protected against attacks, so that no undocumented security critical side effects on the security functions of the TOE are coming from this network.

4.3 Security Objectives Rationale

4.3.1 Security Objective Coverage

The following tables provide a mapping of security objectives to the environment defined by the threats, policies and assumptions, illustrating that each security objective covers at least one threat and that each threat is countered by at least one objective, assumption or policy.

	T.MODIFY	T.DISCLOSE	A.SINGLE	A.PHYSICAL	A.USER	A.CONNECT	A.KEYDIS	A.PASSWORDDIS	P.ALGORITHM
O.MODIFY	X								
O.DISCLOSE		X							
O.ALGORITHM	X	X							X
OE.SINGLE			X						
OE.PHYSICAL		X		X					
OE.USER	X	X			X			X	
OE.PASSWORDDIS	X	X						X	
OE.KEYDIS	X	X					X		
OE.CONNECT						X			

Table 4 Objectives related to threats,assumptions and policies

4.3.2 Security Objectives Sufficiency

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat:

Threat	Is addressed by
T.DISCLOSE	O.DISCLOSE (existence of mechanism to protect confidentiality itself) requires the TOE to provide mechanism of high quality to protect the confidentiality of the file's content while transferring it over any unprotected communication channel. OE.KEYDIS and OE.PASSWORDDIS requires in addition that the TOE only uses keys for encryption / signing and passwords to access private keys of high quality (e.g. exclusion of weak keys and providing a sufficient key length to protect against successful brute-force key search or against attacks together with methods of cryptanalysis). T.DISCLOSE is diminished by O.ALGORITHM reducing the likelihood of a launched attack being successful; greater expertise and greater resources are needed from the attacker to perform attacks based on cryptanalysis. T.DISCLOSE is diminished by restricting potential attackers in opportunities to decrypt the private keys in certificate store. Also OE.KEYDIS requires the keys not to be disclosed to unauthorized users. Thus and OE.USER requiring that users to be trustworthy and well trained restricts the opportunity of unauthorized users possessing the right keys as well. T.DISCLOSE is diminished by O.DISCLOSE together with OE.KEYDIS , OE.PASSWORDDIS and OE.USER. T.DISCLOSE is also diminished by OE.PHYSICAL which requires attacker can not access target machine via unauthorized ways.
T.MODIFY	O.MODIFY(existence of mechanism to detect integrity violations itself) requires the TOE to provide mechanism of high quality to detect integrity violations of the file's content while transferring it over any unprotected communication channel. OE.KEYDIS requires in addition that the TOE only uses keys of high quality for integrity checks. T.MODIFY is diminished by O.ALGORITHM reducing the likelihood of a launched attack being successful; greater expertise and greater resources are needed from the attacker to perform attacks based on cryptanalysis. T.MODIFY is diminished since restricting potentialattackers in opportunities to decrypt the keystore and get access to the key used for validating the integrity of transmitted files. Also OE.KEYDIS is requiring the key used for signing not to be disclosed to unauthorized users. Thus and OE.USER requiring users to be trustworthy and well trained restricts the opportunity of unauthorized users possessing the right key. T.MODIFY is diminished by O.MODIFY together with OE.KEYDIS and OE.USER.

Table 5 Sufficiency of objectives countering threats

The following rationale provides justification that the security objectives for the environment are suitable to cover each individual assumption, that each security objective for the environment that traces back to an assumption about the environment of use of the TOE, when achieved, actually contributes to the environment achieving consistency with the assumption.

Assumption	Is fulfilled by
A.KEYDIS: "It is assumed that keys used for encryption/decryption and as well as the associated keys used for signing are of high quality and are not disclosed to unauthorized users."	OE.KEYDIS requires that keys used for encryption/decryption as well as the keys used for integrity checks must be of high quality and must not be disclosed to unauthorized users. Therefore OE.KEYDIS is only a restatement of A.KEYDIS i.e. OE.KEYDIS fulfils exactly the assumption A.KEYDIS.
A.PASSWORDDIS: "It is assumed that passwords used for accessing private keys in the certificate store or smartcards are of high quality and are not disclosed to unauthorized users."	OE.PASSWORDDIS requires that passwords used for accessing the private keys in the certificate store and smartcards must be of high quality and must not be disclosed to unauthorized users. Therefore OE.PASSWORDDIS is only a restatement of A.PASSWORDDIS i.e. OE.PASSWORDDIS fulfils exactly the assumption A.PASSWORDDIS.
A.SINGLE: "The TOE runs on a single user machine with access protected by the TOE environment; i.e. only authorised users of the TOE environment may access the TOE. This includes access control provided by the operating system or equivalent and protection against malware."	OE.SINGLE requires that the TOE must be run on a single user machine with access to the TOE protected by the TOE environment; i.e., only authorised users of the TOE environment have access to the TOE. This includes access control provided by the operating system and protection against malware. Therefore OE.SINGLE is only a restatement of A.SINGLE; i.e. OE.SINGLE fulfils exactly the assumption A.SINGLE.
A.PHYSICAL: "The TOE is operated in a physically secure and well managed environment."	OE.PHYSICAL requires that the TOE must be run and therefore operated n a physically secure and well managed environment. Therefore OE.PHYSICAL is merely a restatement of A.PHYSICAL; i.e. OE.PHYSICAL fulfils the assumption A.PHYSICAL.
A.USER: "The TOE user is trustworthy and trained to manage and perform encryption of classified information in accordance with any existing security policies and information classification policies. This means especially that he knows how to classify information and how to deal with, e.g., encrypting all files containing sensitive information with the appropriate key before exporting the file out of the TOE and/or its TOE environment."	OE.USER requires that the TOE User is trustworthy and trained to perform all actions in accordance with any existing security policies and information classification policies. OE.USER is merely a restatement of A.USER where the explanation of performing actions in accordance with any existing security policies and information classification policies is not given again because this has to be be clear to the reader. Therefore OE.USER fulfills the assumption A.USER.

A.CONNECT: "The single user PC on which the TOE is running is not connected directly to an untrusted network. This means that the PC is either assumed not to be connected to any networks or it is connected to a trusted network which is protected against attacks, so that no undocumented security critical side effects on the security functions of the TOE, which is resided in the PC, are assumed coming from this network."	OE.CONNECT requires that the single user PC on which the TOE is running must not be connected directly to an untrusted network. This means that the PC must either not be connected to any networks or it must be connected to a trusted network, which is protected against attacks, so that no undocumented security critical side effects on the security functions of the TOE are coming from this network. Therefore OE.CONNECT is merely a restatement of A.CONNECT; i.e. OE.CONNECT fulfils the assumption A.CONNECT.

Table 6 Sufficiency of objectives meeting assumptions

The following rationale provides justification hat the security objectives are suitable to cover each individual organizational security policy, that each security objective that traces back to an OSP, when achieved, actually contributes to the implementation of the OSP, and that if all security objectives that trace back to an OSP are achieved,the OSP is implemented:

OSP	Is addressed by
P.ALGORITHM: "The TOE shall only allow the use of approved encryption algorithms and key lengths."	O.ALGORITHM requires that only approved encryption algorithms and key lengths must be used. Therefore O.ALGORITHM implements exactly the policy P.ALGORITHM.

Table 7 Sufficiency of objectives meeting OSPs

5. EXTENDED COMPONENTS DEFINITIONS

This ST does not define extended components

6. IT SECURITY REQUIREMENTS

The following table gives an overview of the functional components from the Common Criteria Part 2 that are relevant for this TOE.

Component	Component Name
FCS_CKM.1	Cryptographic key generation
FCS_CKM.4	Cryptograhic key destruction
FCS_COP.1	Cryptographic operation
FDP_ACC.1	Subset access control
FDP_ACF.1	Security attribute based access control
FDP_DAU.2	Data Authentication with Identity of Guarantor
FDP_ETC.1	Export of user data without security attributes
FDP_ITC.1	Import of user data without security attributes
FIA_UID.1	Timing of identification
FIA_UAU.1	Timing of authentication
FIA_UAU.7	Protected authentication feedback
FMT_MSA.1	Management of security attributes
FMT_MSA.2	Secure security attributes
FMT_MSA.3	Static attribute initialization
FMT_SMF.1	Specification of Management Functions

Table 8 Functional Requirements on the TOE

The TOE will implement only one Security Function Policy (SFP) called private key access control SFP. The policy's name indicates that it is a policy regulating the access to the private key.
The SFP regulates that when the user wants to access the private key in the certificate store or smartcards he must provide the corresponding password.

6.1 TOE Security Functional Requirements

6.1.1 Class FCS - Cryptographic Support

6.1.1.1 FCS_CKM.1 – Cryptographic key generation (standard key)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[assignment: random],** and specified cryptographic key sizes**,[assignment: 256 bits],** that meet the

following: **[assignment conform to the FIPS 197].**

6.1.1.2 FCS_CKM.4 – Cryptographic key destruction

FCS_CKM.4 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method,**[assignment: zeroization of memory allocated to symmetric keys used in encryption and decryption of files and the private keys just after the file operation is completed. Symmetric keys and private keys are not stored in anywhere else.]** that meet the following:**[assignment: none]**

6.1.1.3 FCS__COP.1/KEYENC – Cryptographic Operation (key encryption)

FCS_COP.1.1 The TSF shall perform **[assignment: encryption of symmetric keys]** in accordance with a specified cryptographic algorithm,**[assignment: RSA]** and cryptographic key sizes **[assignment: 2048]** that meet the following:**[assignment: conform to the FIPS 140-2 Annex.A.]**

6.1.1.4 FCS__COP.1/KEYDEC – Cryptographic Operation (key decryption)

FCS_COP.1.1 The TSF shall perform **[assignment: decryption of symmetric keys]** in accordance with a specified cryptographic algorithm,**[assignment: RSA]** and cryptographic key sizes **[assignment: 2048]** that meet the following:**[assignment: conform to the FIPS 140-2 Annex.A.]**

6.1.1.5 FCS__COP.1/ENC – Cryptographic Operation (file encryption)

FCS_COP.1.1 The TSF shall perform **[assignment: encryption of files]** in accordance with a specified cryptographic algorithm,**[assignment: AES]** and cryptographic key sizes **[assignment: 256]** that meet the following:**[assignment: conform to the FIPS 197.]**

6.1.1.6 FCS__COP.1/DEC – Cryptographic Operation (file decryption)

FCS_COP.1.1 The TSF shall perform **[assignment: decryption of encrypted files]** in accordance with a specified cryptographic algorithm**[assignment: AES]** and cryptographic key sizes **[assignment: 256** that meet the following:**[assignment: conform to the FIPS 197].**

6.1.1.7 FCS__COP.1/SIGN – Cryptographic Operation (file signing)

FCS_COP.1.1 The TSF shall perform **[assignment: signing files]** in accordance with a specified cryptographic algorithm,**[assignment: RSA]** and cryptographic key sizes,**[assignment: 2048]** that meet the following:**[assignment: conform to the RFC 4056]**.

6.1.1.8 FCS__COP.1/VERIFY – Cryptographic Operation (file verification)

FCS_COP.1.1 The TSF shall perform **[assignment: verification of signed files]** in accordance with a specified cryptographic algorithm **[assignment: RSA]** and cryptographic key sizes **[assignment: 2048]** that meet the following: **[assignment:conform to the RFC 4056]**.

6.1.2 Class FDP – User Data Protection

6.1.2.1 FDP_ACC.1 – Subset access control (Private Key Access Control)

FDP_ACC.1.1 The TSF shall enforce **[assignment: the private key access control SFP]** on **[assignment: all users, The private keys in the certificate store and in the smartcards.]**

6.1.2.2 FDP_ACF.1 – Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce **[assigment: the private key access control SFP]** to objects based on the following: **[assignment: users and the password.]**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
[assignment:

- **the user must enter the correct password to access the private keys in the certificate store or smartcard.]**

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

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

6.1.2.3 FDP_ITC.1 - Import of user data without security attributes

FDP_ITC.1.1 The TSF shall enforce the [**assignment: the private key access control SFP**] when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the userdata when imported from outside the TOE.

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

6.1.2.4 FDP_ETC.1 Export of user data without security attributes

FDP_ETC.1.1 The TSF shall enforce the [**assignment: the private key access control SFP**] when exporting user data, controlled under the SFP(s), outside of the TOE.

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

6.1.2.5 FDP_DAU.2 –Data Authentication with Identity of Guarantor

FDP_DAU.2.1 The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of **[assignment: signed and encrypted files.]**

FDP_DAU.2.2 The TSF shall provide **[assignment: users]** with the ability to verify evidence of the validity of the indicated information and the identity of the user that generated the evidence.

6.1.3 Class FIA – Identification and Authentication

6.1.3.1 FIA_UID.1 – Timing of Identification

FIA_UID.1.1 The TSF shall allow [**assignment: any operations except accessing to the private keys**] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

6.1.3.2 FIA_UAU.1 Timing of authentication

FIA_UAU.1.1 The TSF shall allow **[assignment: any operations except accessing to the private keys]** on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

6.1.3.3 FIA_UAU.7 Protected authentication feedback

FIA_UAU.7.1 The TSF shall provide only **[assignment: *dots as digits of the password*]** to the user while the authentication is in progress.

6.1.4 Class FMT – Security Management

6.1.4.1 FMT_MSA.1 – Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the **[assignment: private key access control SFP]** to restrict the ability to **[selection: modify]** the security attributes **[assignment: certificate store password]** to **[assignment: any user who knows the actual password.]**

Application Note: The TOE is not aware of any user roles but controls the access to the certificate store and smartcard via a password. The TOE is assumed to operate on a single user machine with only one user having access to the TOE.

6.1.4.2 FMT_MSA.2 – Secure security attributes

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes **[assignment: symmetric keys]**

Application Note: Only symmetric keys with a key length of 256 bits and with algorithm attributes of cryptographic algorithms supported by the TOE are generated and used by the TOE.

6.1.4.3 FMT_MSA.3 – Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the **[assignment: private key access control SFP]** to provide **[selection: no]** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow **[assignment: any user who knows the certificate store password]** to specify alternative initial values to override the default values when an object or information is created.

Application Note: There are no default values for certirficate store passwords; the user has to choose a password himself starting the application the first time.

6.1.4.4 FMT_SMF.1 – Specification of Management Functions

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

- **change certificate store password**

- **export private keys (only soft keys)**
- **import private keys (only soft keys)]**

Application Note: All management functions except delete private key requires the user to present the password of the certificate store.

6.2 TOE Security Assurance Requirements

The target assurance components for this TOE are those for EAL4+ (ALC_FLR.2) as specified in Part 3 of the CC. The following table provides an overview of the assurance components that form the assurance level for the TOE.

Assurance class	Assurance components
ADV:Development	ADV_ARC.1 Security architecture description
	ADV_FSP.4 Complete functional specification
	ADV_IMP.1 Implementation representation of the TSF
	ADV_TDS.3 Basic Modular design
AGD:Guidance and Documentation	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC:Life cycle support	ALC_CMC.4 Production support, acceptance procedures and automation
	ALC_CMS.4 Problem tracking CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_LCD.1 Developer defined life-cycle model
	ALC_TAT.1 Well-defined development tools
	ALC_FLR.2 Flaw reporting procedures
ASE:Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definitions
	ASE_INT.1 ST Introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE:Tests	ATE_COV.2 Analysis of coverage

	ATE_DPT.1 Testing: high-level design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
AVA:Vulnerability assessment	AVA_VAN.3 Focused Vulnerability analysis

Table 9 Security Assurance Components

6.3 Security Requirements Rationale

6.3.1 Security Requirements Coverage

The following tables provide a mapping of the relationships of security functional requirements to objectives, illustrating that each security requirement covers at least one objective and that each objective is covered by at least one security requirement.

	O.MODIFY	O.DISCLOSE	O.ALGORITHM
FCS_CKM.1	X	X	
FCS_CKM.4		X	
FCS_COP.1/KEYENC		X	X
FCS_COP.1/KEYDEC		X	X
FCS_COP.1/ENC		X	X
FCS_COP.1/DEC		X	X
FCS_COP.1/SIGN	X		X
FCS_COP.1/VERIFY	X		X
FDP_ACC.1	X	X	
FDP_ACF.1	X	X	
FDP_DAU.2	X		X
FDP_ETC.1		X	
FDP_ITC.1		X	
FIA_UID.1	X	X	
FIA_UAU.1	X	X	
FIA_UAU.7		X	
FMT_MSA.1	X	X	
FMT_MSA.2			X
FMT_MSA.3			X
FMT_SMF.1			X

Table 10 TOE Security objectives meeting SFRs

6.3.2 Functional Security Requirements Sufficiency

Objective	Is fulfilled by the SFRs
O.MODIFY	The mechanisms to detect loss of integrity of the information included in a transmitted file is achieved by the cryptographic operations FCS_COP.1/SIGN together with FCS_COP.1/VERIFY. On the sender side the data is signed (FCS_COP.1/SIGN) first, before it could be verified on the receiver side (FCS_COP.1/VERIFY) where the properly detection of potential integrity violation takes place. FDP_DAU.2.1 and FDP_DAU.2.2 ensures that transferred data is digitally signed and verified upon receival. With FMT_MSA.1 only users who know the password could change it. FMT_MSA.2 ensures that only keyed checksum functions and key length are generated and used that are supported by the TOE. FMT_SMF.1 provides the specific management functions for key generation. FIA._UID.1 and FIA_UAU.1 ensures that only authenticated user have access to signing private keys and authentication feedback is protected.

O.DISCLOSE	The mechanism to protect files during transmission against confidentiality violation is achieved by the cryptographic operation FCS_COP.1/KEYENC and FCS_COP.1/ENC ensuring, that the files encrypted such that the content is confidentiality protected and that only parties who has the privaye asymmetric key of one of the recipients could decrypt the file. The cryptographic operation FCS_COP.1/DEC and FCS_COP.1/KEYDEC providing the file decryption on the receivers site is strictly spoken not needed to achieve integrity protection but implemented to provide access to the encrypted information to those who are authorized. These mechanisms are using keys, therefore supporting requirements regarding keys are needed. Symmetric encryption keys are generated (FCS_CKM.1) on the sender side, and encrypted asymmetrically to the private key on the receiver side. The encryption key used for the private key decryption in the certificate store on the receiver side is derived from a password which has to be assigned first to the certificate store during creation on the receiver side. FCS_CKM.1 is needed. FCS_CKM.4 ensures that all symmetric keys used for content encryption is destructed just after their use. With FMT_MSA.1 only users who know the password could change it. FMT_MSA.2 ensure that only the approved algorithms for encryption/decryption and key length are generated and used that are supported by the TOE. FMT_SMF.1 provides the specific management functions for key generation. FIA._UID.1 and FIA_UAU.1 ensures that anly authenticated user have access to key-encryption private keys. FIA_UAU.7 ensures that authentication feedback is protected. FDP_ETC.1 and FDP_ITC.1 ensures that the private keys of users are imported into and exported from the certificate store.

O.ALGORITHM	The key generation and derivation requirement achieves that only approved key generation and derivation algorithms with a specified key size are allowed (FCS_CKM.1). Further only approved algorithms with specified key sizes are allowed for cryptographic operations in the cryptographic operation requirements (FCS_COP.1/KEYENC,FCS_COP.1/KEYDEC FCS_COP.1/ENC,FCS_COP.1/DEC, FCS_COP.1/SIGN, FCS_COP.1/VERIFY). These algorithms could not be managed by the user (see FMT_SMF.1 – no management function exists) they are fixed. FMT_MSA..2 ensures that only the approved algorithms for encryption and decryption as well as key length are accepted by the TOE. FMT_MSA.3 ensures that no insecure algorithms and keys are in the keystore unless they have been generated or exported. Thus leads to the fact that only approved encryption algorithms and key length are allowed and used by the TOE.

Table 11 TOE Security Objectives and the Rationale for Mapping to the SFRs

As stated in the tables above, every objective is addressed by at least one security functional requirement and every SFR is necessitated to cover at least one objective. By showing that the stated security objectives are met, we are able to demonstrate the suitability and sufficiency of the chosen SFRs.

6.3.3 Rationale of Selected Assurance Level

The assurance level EAL4+ (ALC_FLR.2) has been chosen as appropriate for an application that is encrypting files in a secure and well managed environment. The attacker is also assumed only to attack the data exported or imported into the TOE and not the TOE itself, thereby limiting the opportunity of an attacker. For these reasons EAL4+ (ALC_FLR.2) is considered a sufficient level of assurance.

6.3.4 Security Requirements Dependency Analysis

Following the Common Criteria and choosing security requirements to be met by a TOE, certain dependencies on other security requirements may arise. The following section shows whether these dependencies are resolved and, in case they are not, gives reasons for that.

6.3.4.1 Security Functional Requirements Dependency Analysis

SFR	Dependencies	Support of the Dependencies
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation], FCS_CKM.4 Cryptographic key destruction,	Fulfilled by FCS_COP.1/ENC Fulfilled by FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	Fulfilled by FCS_CKM.1,
FCS_COP.1/KEYENC	[FDP_ITC.1 Import of user data without security attributes, FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_CKM.1, Fulfilled by FCS_CKM.4
FCS_COP.1/KEYDEC	[FDP_ITC.1 Import of user data without security attributes, FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_CKM.1, Fulfilled by FCS_CKM.4
FCS_COP.1/ENC	[FDP_ITC.1 Import of user data without security attributes, FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_CKM.1, Fulfilled by FCS_CKM.4
FCS_COP.1/DEC	[FDP_ITC.1 Import of user data without security attributes, FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_CKM.1, Fulfilled by FCS_CKM.4
FCS_COP.1/SIGN	[FDP_ITC.1 Import of user data without security attributes, FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_CKM.1, Fulfilled by FCS_CKM.4
FCS_COP.1/VERIFY	[FDP_ITC.1 Import of user data without security attributes, FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_CKM.1, Fulfilled by FCS_CKM.4
FDP_ACC.1	FDP_ACF.1 Security attribute based access control	Fulfilled by FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialization	Fulfilled by FDP_ACC.1, Fulfilled by FMT_MSA.3

FDP_DAU.2	FIA_UID.1 Timing of Identification	Fullfilled by FIA_UID.1
FDP_ETC.1	FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control	Fulfilled by FDP_ACC.1
FDP_ITC.1	[FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialisation	Fulfilled by FDP_ACC.1 Fulfilled by FMT_MSA.3
FIA_UID.1	No dependencies	n.a.
FIA_UAU.1	FIA_UID.1 Timing of identification	Fulfilled by FIA_UID.1
FIA_UAU.7	FIA_UAU.1 Timing of authentication	Fulfilled by FIA_UAU.7
FMT_MSA.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	Fulfilled by FDP_ACC.1 Fulfilled by FDP_SMF.1 The lack of FMT_SMR.1 is justified below this table.
FMT_MSA.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FDP_ACC.1 Fulfilled by FMT_MSA.1 The lack of FMT_SMR.1 is justified below this table.
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	Fulfilled by FMT_MSA.1 The lack of FMT_SMR.1 is justified below this table.
FMT_SMF.1	No dependencies	n.a.

Table 12 Dependencies between the SFR for the TOE

Justification of lack of FMT_SMR.1:

It is assumed that TOE runs in a secure environment with one type of user and there are no roles. Only authorized users can access the TOE and this authorization is fulfilled by the operating system.

6.3.4.2 Security Assurance Dependencies Analysis

The assurance level selected within this TOE is EAL4+(ALC_FLR.2). Since the dependency analysis for EAL4+(ALC_FLR.2) has been performed by the authors of the CC and as all dependent assurance components have been included, all dependencies of the assurance components within this Security Target are resolved.

6.3.5 Internal Consistency and Mutual Support of SFRs

Section 6.3.2 has already demonstrated how the IT security requirements work together to implement the individual objectives for the TOE and the IT environment. This section will elaborate on the internal consistency and mutual support of the IT security requirements.

The TOE's purpose is to enable users to exchange electronic documents securely over unprotected communication paths by ensuring confidentiality with encryption and the detection of loss of integrity by using electronic signature.
Therefore cryptographic keys have to be generated first (FCS_CKM.1). They are in smartcards or in the hard disk within a password protected store file.

When using Kermen Portable for the first time an initial password for the certificate store has to be set by the user. From this password the key for encryption/decryption of the private keys stored in

the certificate store is derived. Therefore the private keys used in decryption or signing are encrypted with the derived encryption key (FCS_COP.1/KEYENC).

Every further time the user starts Kermen Portable and tries to decrypt an encrypted file, if he has private key in the certificate store, he has to decrypt the private key first. Therefore he has to enter the actual password of the certificate store. The keys are derived by PKCS12PBKDF2 key derivation algorithm (FCS_CKM.1). The user could perform some management functions on the certificate store , such as importing new private keys or exporting existing keys only if he provide the correct password.(FMT_SMF.1.) The certificate store passwords could be changed. When private keys are stored in smartcards, the security of the keys are provided by the smartcards. Private keys are only accessible after the smartcard password is provided and it is impossible to extract private keys from the smartcards.

When the user wants to create an encrypted file for other user, he uses the public key of that user placed in his certficate. Before the file is encrypted, it is signed by the users' signing private key placed in his smartcard or the certificate store, by providing the required password. The signature protects the file against integrity attacks and corruptions. After signed-and-encrypted file is created, it can be transported in insecure environments. The recipient of the encrypted file can decrypt it by using his private key as explained previously. Then he verifies the signature of the sender and accesses the plain file.

7. TOE SUMMARY SPECIFICATION

The TOE summary specification provides a complete high-level definition of the security functions and assurance measures of the TOE and their relationship to the security functional and assurance requirements of this ST.
The TOE summary specification identifies the security functions that the TOE implements to meet the requirements defined in chapter 5 of the security target.

7.1 TOE Security Functions

This chapter describes the IT security functions of the TOE and their relation to the security functional requirements which they are supposed to meet.
A mapping of security functions against requirements is provided in clause 7.2 of the rationale part.

7.1.1 SF.KEYGEN- Key Generation

Symmetric encryption keys are created randomly by using the seed provided by smartcards (if applicable).

7.1.2 SF.FILE_CRYPT - File encryption / decryption

Data files are encrypted and decrypted using the CMS data encryption standard (enveloped data) including asymmetric encryption of symmetric keys with RSA and symmetric encryption of the data with AES algorithm in CBC mode with a 256-bit key. The format of the encrypted data in both cases is based on EnvelopedData format [RFC 5652].

7.1.3 SF.FILE_SIGN – File signing

Encrypted Data files are signed and verified using the CMS data signature standard (Signed Data) including asymmetric signature of data with RSA in order to ensure that the integrity of files are preserved. The format of the Signed Data in both cases is based on SignedData format [RFC 5652].

7.1.4 SF.CHECK_INTEGRITY – File verification

The first step is to validate that the file is in CMS SignedData format Afterwards the integrity check is implemented as stated in RFC 5652. Only if SF.CHECK_INTEGRITY is successful, decryption according to SF.FILE_CRYPT is performed with the corresponding private key.

7.1.5 SF.MANAGE

The TOE allows the user to perform the following management functions:

- delete private key (only soft keys)
- change certificate store password
- export private keys (only soft keys)
- import private keys (only soft keys)

All management functions except the function deleting the private key requires the user to present the correct password of the certificate store. The default certificate store can be deleted before the password is given and SF.CHECK_INTEGRITY, SF.KEYGEN and SF.FILE_CRYPT (keystore decryption) is performed.

For export of private keys, the keys are placed in a PFX file and a key unwrap password must be assigned by the user. For import of private keys, the key unwrap password of the imported PFX file must be entered by the user.
For changing the certificate password the user is called to enter the current password first.

7.1.6 SF.CLEAR

Encryption keys and soft private keys (private keys provided by PFX files) are cleared from the memory just after their use and the memory locations allocated for them are zeroized.

7.2 TOE Summary Specification Rationale

The TOE IT security functions work together to satisfy the security functional requirements. Below a justification is presented for each SFR, how the related security functions meet the requirements, and as well for the sum of SARs.
By examining the TOE summary specification and this justification carefully, it becomes clear that the security functions are a well defined set combined to build a sound application for file encryption and therefore to meet the requirements defined
in this ST.
The following tables provide a mapping between security functions and security functional requirements.

	SF.KEYGEN	SF.FILE_CRYPT	SF.FILE_SIGN	SF.CHECK_INTEGRITY	SF.MANAGE	SF.CLEAR
FCS_CKM.1	X					
FCS_CKM.4						X
FCS_COP.1/ENC		X			X	
FCS_COP.1/DEC		X			X	
FCS_COP.1/SIGN			X			
FCS_COP.1/VERIFY				X		
FDP_ACC.1		X	X		X	
FDP_ACF.1		X	X		X	
FDP_DAU.2			X	X		
FDP_ETC.1					X	
FDP_ITC.1					X	
FIA_UID.1		X	X		X	
FIA_UAU.1		X	X		X	
FIA_UAU.7		X	X		X	
FMT_MSA.1					X	
FMT_MSA.2	X	X			X	
FMT_MSA.3					X	
FMT_SMF.1					X	

Table 13 TOE Security Functions meeting SFRs and Vice Versa

7.2.1 Security Functions Justifications

The following table shows that the IT security functions (SF) as specified in the TOE Summary Specification meet all the security functional requirements (SFR) for the TOE and work together to satisfy the TOE security functional requirements.

SFR	Security Functions(TOE Summary Specification)
FCS_CKM.1	The requirement for key generation is satisfied by the security function SF.KEYGEN, which will generate all symmetric encryption keys (AES Keys).
FCS_CKM.4	The requirement for key desytruction is satisfied by the security function SF.CLEAR which states that all encryption keys and soft private keys are zeroized and cleared from memory..

FCS_COP.1	The requirement for file encryption is satisfied by the security function SF.FILE_CRYPT, which specifies that asymmetric encryption algorithm specified in user certificates is used for the encryption of symmetric keys and AES in CBC mode with a 256 bit is used for data encryption, conforming to [RFC 5652].The requirement for the file signing is satisfied by the security function SF.FILE_CHECK_INTEGRITY and SF_FILE_SIGN which specifies the hash and signature algorithms are used in the file signing conforming to [RFC 5652].
FDP_ACC.1	The requirement for access control is satisfied by the management function SF.MANAGE, implementing the private key access control SFP for all users accessing the private key. Moreover, the requirement for access control is also satisfied by SF.FILE_CRYPT and SF.FILE_SIGN security functions implementing the private key access control SFP for all users accessing the private key.
FDP_ACF.1	The requirement for access control rules is satisfied by the management function SF.MANAGE, implementing that users must present the correct certificate store password before performing the management functions, except the function deleting the default keystore, this can be performed without entering password as a emergency erase functionality. Moreover, the requirement for access control rules is also satisfied by SF.FILE_CRYPT and SF.FILE_SIGN security functions implementing that users must present the correct certificate store password before accessing the priavet keys.
FDP_DAU.2	The requirement for data authentication with identity of guarantor is satisfied by the security function SF_CHECK_INTEGRITY and SF_FILE_SIGN which implements signing and verification of encrypted files
FDP_ETC.1	The requirement for export of user data without security attributes is satisfied by the security function SF_MANAGE, which allows export of private keys from the certificate store.
FDP_ITC.1	The requirement for import of user data without security attributes is satisfied by the security function SF_MANAGE, which allows import of private keys into the certificate store.
FIA_UID.1	The requirement for timing of identification is satisfied by the security function SF.FILE_CRYPT and SF_FILE_SIGN implementing the private key access control SFP for all users accessing the private key.
FIA_UAU.1	The requirement for timing of authentication is satisfied by the security function SF.FILE_CRYPT and SF_FILE_SIGN implementing the private key access control SFP for all users accessing the private key.

FIA_UAU.7	The requirement for protection of authentication is satisfied the security function SF.FILE_CRYPT and SF_FILE_SIGN showing only dots as the digits of passwords when the users trying to authenticate to access their private keys.
FMT_MSA.1	The requirement for authorization of password changes for the certificate store is implemented in SF.MANAGE, allowing password changes only to those users who know the current password of the certificate store.
FMT_MSA.2	The requirement for secure security attributes is satisfied by SF.KEYGEN together SF.FILECRYPT. SF.KEYGEN ensures that only secure values are generated SF.FILECRYPT works with only secure attributes (key sizes). Therefore only secure attributes could be used within Kermen Portable.
FMT_MSA.3	The requirement for static attribute installation is fulfilled by SF.MANAGE, by not providing any default attributes that could be insecure.
FMT_SMF.1	The requirement for the TSF to provide management functions is satisfied by SF.MANAGE.

7.2.2 Mutual Support of Security Functions

The IT security functions provided by the TOE work together to satisfy the TOE security functional requirements defined in this Security Target. The tight relationship between the defined requirements and the fulfilment of these requirements by security functions, as illustrated above in section 7.2.1, provides no room for the introduction of potential security weaknesses not identified in this document.

7.2.3 Assurance Measures Rationale

Since TOE is defined on the information security domain,a valunerability analysis assuming an attack potential of at least Enhanced-Basic is considered to be required. Therefore EAL4+(ALC_FLR.2) is chosen which includes AVA_VAN.3

8. APPENDIX

A.1 Abbreviations

CCMB	Common Criteria Maintenance Board
EAL	Evaluation Assurance Level
GUI	Graphical User Interface
IT	Information Technology
PP	Protection Profile
RFC	Request for comments
SF	Security Function
SFP	Security Function Policy
SOF	Strength of Function
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy
CMS	Cryptographic Message Syntax