

INFORMATICS AND INFORMATION SECURITY RESEARCH CENTER

NATIONAL RESEARCH INSTITUTE OF ELECTRONIC AND CRYPTOLOGY

**KERMEN v1.6**

**Security Target Lite**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
| Title: | Kermen Security Target Lite Document | | |
| Author: | Metin SIMSEK  Nur MERDANOGLU  Yusuf DUNDAR | Status: | Release |
| Version: | 1.0 | Classification: | Public |
| FileName: | KERMEN\_ST\_Lite.pdf | Date: 14.10.2020 |  |

**DOCUMENT HISTORY**

|  |  |  |
| --- | --- | --- |
| **Version** | **Changes To the Previous Version** | **Date** |
| 1.0 | First Publication | 14.10.2020 |

TABLE OF CONTENTS

[INDEX OF TABLES 5](#_Toc38289295)

[FIGURE INDEX 5](#_Toc38289296)

[1. INTRODUCTION 6](#_Toc38289297)

[1.1 ST Reference 6](#_Toc38289298)

[1.2 TOE Reference 6](#_Toc38289299)

[1.3 TOE Overview 6](#_Toc38289300)

[1.3.1 TOE Definition 6](#_Toc38289301)

[1.3.2 TOE usage and security features for operational use 6](#_Toc38289303)

[1.3.3 TOE major security functions 6](#_Toc38289304)

[1.3.4 Non-TOE hardware/software/firmware required by the TOE 7](#_Toc38289305)

[1.4 TOE Description 7](#_Toc38289306)

[1.4.1 Introduction 7](#_Toc38289307)

[1.4.2 TOE Definition Scope 8](#_Toc38289308)

[1.4.3 Supported Platforms and Environment 9](#_Toc38289309)

[1.4.4 Installation 9](#_Toc38289310)

[1.4.5 Configuration 10](#_Toc38289311)

[1.4.6 TOE Operation and Use 10](#_Toc38289312)

[1.4.7 TOE Environment and Physical Protection 11](#_Toc38289313)

[1.5 Related Standards and Documents 11](#_Toc38289314)

[2. Conformance Claim 12](#_Toc38289315)

[2.1 CC Conformance Claim 12](#_Toc38289316)

[2.2 PP Claim 12](#_Toc38289317)

[2.3 Package Claim 12](#_Toc38289318)

[2.4 Conformance Rationale 13](#_Toc38289319)

[SecurIty Problem DEFINITION Envıronment 14](#_Toc38289320)

[2.5 Secure Usage Assumptions 14](#_Toc38289321)

[2.6 Organizational Security Policies 14](#_Toc38289322)

[2.7 Threats 15](#_Toc38289323)

[2.7.1 Assets and Agents 15](#_Toc38289324)

[2.7.2 Threats addressed by the TOE and the TOE environment 16](#_Toc38289325)

[3. SecurIty ObjectIves 18](#_Toc38289326)

[3.1 Security Objectives for The TOE 18](#_Toc38289327)

[3.2 Security Objectives for the IT and non-IT Environment 18](#_Toc38289328)

[3.3 Security Objectives Rationale 19](#_Toc38289329)

[3.3.1 Security Objective Coverage 19](#_Toc38289330)

[3.3.2 Security Objectives Sufficiency 19](#_Toc38289331)

[4. Extended components defınıtıons 22](#_Toc38289332)

[5. IT Securıty REQUIREMENTS 22](#_Toc38289333)

[5.1 Operation Notation for Functional Requirements 23](#_Toc38289334)

[5.2 TOE Security Functional Requirements 24](#_Toc38289335)

[5.2.1 Class FCS - Cryptographic Support 24](#_Toc38289336)

[5.2.2 Class FDP – User Data Protection 26](#_Toc38289337)

[5.2.3 Class FIA – Identification and Authentication 28](#_Toc38289338)

[5.2.4 Class FMT – Security Management 29](#_Toc38289339)

[5.3 TOE Security Assurance Requirements 30](#_Toc38289340)

[5.4 Security Requirements Rationale 31](#_Toc38289341)

[5.4.1 Security Requirements Coverage 31](#_Toc38289342)

[5.4.2 Functional Security Requirements Sufficiency 32](#_Toc38289343)

[5.4.3 Rationale of Selected Assurance Level 34](#_Toc38289344)

[5.4.4 Security Requirements Dependency Analysis 34](#_Toc38289345)

[5.4.5 Internal Consistency and Mutual Support of SFRs 37](#_Toc38289346)

[6. TOE Summary Specıfıcatıon 38](#_Toc38289347)

[6.1 TOE Security Functions 38](#_Toc38289348)

[6.1.1 SF.KEYGEN - Key Generation 38](#_Toc38289349)

[6.1.2 SF.FILE\_CRYPT - File encryption / decryption 38](#_Toc38289350)

[6.1.3 SF. MAIL\_CRYPT - Mail encryption/decryption 38](#_Toc38289351)

[6.1.4 SF.FILE\_SIGN – File signing 38](#_Toc38289352)

[6.1.5 SF.MAIL\_SIGN – Mail signing 38](#_Toc38289353)

[6.1.6 SF.CHECK\_FILE\_INTEGRITY – File verification 38](#_Toc38289354)

[6.1.7 SF.CHECK\_MAIL\_INTEGRITY – Mail verification 39](#_Toc38289355)

[6.1.8 SF.MANAGE 39](#_Toc38289356)

[6.1.9 SF.CLEAR 39](#_Toc38289357)

[6.2 TOE Summary Specification Rationale 39](#_Toc38289358)

[6.2.1 Security Functions Justifications 40](#_Toc38289359)

[6.2.2 Mutual Support of Security Functions 42](#_Toc38289360)

[6.2.3 Assurance Measures Rationale 42](#_Toc38289361)

[7. Appendıx 43](#_Toc38289362)

[A.1 Abbreviations 43](#_Toc38289363)

[A.2 References 44](#_Toc38289364)

# INDEX OF TABLES

[Table 1: Assets 15](#_Toc38289365)

[Table 2: Agents 15](#_Toc38289366)

[Table 3 Threats addressed by the TOE 16](#_Toc38289367)

[Table 4 Objectives related to threats, assumptions and policies 19](#_Toc38289368)

[Table 5 Sufficiency of objectives countering threats 20](#_Toc38289369)

[Table 6 Sufficiency of objectives meeting assumptions 22](#_Toc38289370)

[Table 7 Sufficiency of objectives meeting OSPs 22](#_Toc38289371)

[Table 8 Functional Requirements on the TOE 23](#_Toc38289372)

[Table 9 Security Assurance Components 30](#_Toc38289373)

[Table 10 TOE Security objectives meeting SFRs 31](#_Toc38289374)

[Table 11 TOE Security Objectives and the Rationale for Mapping to the SFRs 33](#_Toc38289375)

[Table 12 Dependencies between the SFR for the TOE 37](#_Toc38289376)

[Table 13 TOE Security Functions meeting SFRs and Vice Versa 40](#_Toc38289377)

# FIGURE INDEX

[Figure 1 TOE Architecture and Boundaries 9](#_Toc36829674)

# INTRODUCTION

## ST Reference

|  |  |
| --- | --- |
| **ST Title:** | KERMEN v1.6 Security Target Lite |
| **Product Name:** | KERMEN |
| **Product Version:** | 1.6 |
| **Assurance Level:** | EAL4+ (ALC\_FLR.2) |
| **CC Version:** | **3.1 R5** |
|  |  |
|  |  |
| **ST Version:** | 1. 0 |
| **Key Words:** | File & Email Content Encryption / Decryption |

## TOE Reference

**TOE Identification:** KERMEN Version 1.6

## TOE Overview

### TOE Definition

KERMEN is a data encryption/decryption application intended to protect files and emails. Term “Data” will be used time to time interchangebly in place of any user files and email contents subject to TOE encryption process for the rest of this document.

KERMEN does file encryption/decryption and other related operations on Microsoft Windows platforms and Pardus Operating Systems, email encryption on MS Outlook and Mozilla ThunderBird. The underlying platform for the evaluation is limited to MS Office Outlook Email clients, MS Windows 7 and above operating systems.

FIPS approved open source cryptographic library cryptoPP and OpenSSL are used to implement all cryptographic functions.

### TOE usage and security features for operational use

KERMEN is used to secure data in the form of files and/or emails. Data can be any kind of confidential material. KERMEN encrypts data using user certificates based on RFC 5652 standard. The encrypted data can only be decrypted by addressed users private keys. Encrypted data might be signed to protect integrity of data. Thus, sensitive and confidential data can be protected against unauthorized access with ensured integrity.

### TOE major security functions

KERMEN provided following security features:

* **Encryption of files:** Files are encrypted using the EnvelopedData method defined in [RFC 5652] [[1]](#_A.2_References)
* **Integrity protection of files:** To detect the lossof integrity of data files, encrypted files are signed as described in [RFC5652**]**
* **Encryption of emails:** Emails are encrypted using the S/MIME standard defined in [RFC 5751][[2]](#_A.2_References)
* **Integrity protection of emails:** To detect the lossof integrity of emails, encrypted emails are signed using the S/MIME standard defined in [RFC 5751].
* **Decryption of data:** Encrypted data is decrypted if the user posseses the corresponding private key in his smart card or certificate store.
* **Integrity Check of data:** Validation of the encrypted data indicates that integrity of the original data is ensured.
* **Nonrepudiation of data:** User can claim neither the data is not signed by himself nor it is modified after signing by the nature of the signature operation.
* **Key Storage:** Soft private keys are stored in the certificate store in password based encrypted form.
* **Import/Export Keys:** Certificates and associated private keys can be imported into and exported from the certificate store by using Kermen DEPO GUI.

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

KERMEN requires Windows 7 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. MS Outlook needs to be installed to send encrypted emails. The minimum system requirements for the target machine;

* Intel Core i3 processor
* 3 GB memory
* 20 GB harddisk space
* MS Outlook 2010 and above
* Windows 7 and above

## 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.

### Introduction

KERMEN is a software only product for data encryption/decryption in both Microsoft Windows and Pardus environments, running on a single user PC. The program requires X.509 certificates in order to perform cryptographic operations.

KERMEN implements asymmetric encryption by using EnvelopedData format defined in [RFC 5652] to ensure the confidentiality of the information. The confidential data is signed before and after encryption in order to ensure integrity of data in accordance with the CMS standard defined in [RFC 5652]. Encrypted and signed data can be decrypted and signatures are validated by KERMEN as specified by the the standard.

X.509 certificates are used to identify individuals and perform asymmetric encryption. KERMEN checks the validity of the certificates upon encryption and decryption operations. Certificate chain, referenced root certificates to be included in the certificate store and other structural controls are all done and checked along with validity checks. TOE only works with smartcards conforming to thePKCS #11 standard and have valid international security certifications. The trusted certificates in the certificate store have to be signed by a trusted authority. Only the certificates signed by a trusted authority will be accepted as trusted certificates. Certificate validation is performed according to the steps described in [RFC 5280]. [[3]](#_A.2_References)

KERMEN stores the X.509 user certificates in a custom build certificate store. The private keys can be either in smartcards or in the certificate store in encrypted form by user defined passwords in which case the user is responsible to keep the password secure.

KERMEN provides a utility tool to view, import, export and manage any stored certificates in the certificate store.

KERMEN uses secure erase method to delete confidential files by overwriting files with zeros, ones, and random blocks as described in [NISP SP 800-88]. [[4]](#_A.2_References)

### TOE Definition Scope

The target of evaluation is limited to the software application KERMEN, version 1.6, developed by TÜBİTAK BİLGEM. KERMEN consist of several applications and modules in the form of desktop GUI, shell extension and addin:

* KERMEN SUR is a GUI application for file encryption/decryption.
* KERMEN DEPO is the certificate store database to hold user certificates, keys and CRLs locally. The database is signed and encrypted.
* KERMEN DEPO Goruntuleyici is a desktop GUI application to view and manage certificate store and the underlying API responsible for all of the cryptographic work. API depends on open source Crypto++ and OpenSLL crypto libraries.
* Secure Folder Module (GDM) is a desktop GUI application to manage any number of user folders that their content to be automatically get encrypted and signed.
* KERMEN Desktop Security Module (MGM) is a desktop GUI application comes with a Shell extension for the Windows Context Menu. It lets users do various cryptographic operations such as encyrption/decryption, document signing, certificate and CRLs management, secure file deletion etc.
* KERMEN Sifreci is a desktop GUI application to allow encryption/decryption operations on group of files.
* KERMEN Secure e-Mail Module (EEPM) is an MS Outlook Add-In that intercepts ongoing emails and if classified as confidential and above encrypts the body in S/MIME format. Public keys used for encryption can be stored in LDAP servers and retrieved when needed. Confidential emails are signed by default, confidential data with classification above confidential will also be encrypted. Decryption process does not need KERMEN and handled by MS Outlook alone.

All the cryptographic and related operations are performed via the underlying KERMEN API. Therefore, the GUI components is not security critical. In the following picture the architecture of TOE KERMEN and its boundaries are shown.



Figure 1 TOE Architecture and Boundaries

The physical scope of the TOE includes Application and API files and Dlls, user’s encrypted data and log files. All these files and data are assumed to be protected by their physical environment and by the underlying operating system.

Delivery Method:

TOE is delivered within a CD tagged with TOE reference and components. The CD will be delivered to the recipient with a delivery report. Classification of CD is restricted and hash is provided to recipient in order to check the content. CD includes:

TOE

Requirement Package

Institution Setting Package

TOE User Guide and educational documents

### Supported Platforms and Environment

Although KERMEN supports Pardus Operating System and Mozilla ThunderBird, the underlying platform for the evaluation is limited to the MS Windows 7 and above operating systems with MS Outlook. MS Outlook is required to be able to install and use EEPM. To be able to use smart cards a card reader is required. No additional special equipment or infrastructure is needed.

### Installation

No installation required in order to run KERMEN SUR and Kermen Depo Goruntuleyici. Only copying program files is sufficient to use them in portable mode with a memory stick.

KERMEN MGM, KERMEN GDM, KERMEN Sifreci and KERMEN EEPM must have three seperate setup packages to be installed to function properly. The first package called “Requirements Package” installs infrastructure libraries. The second package called “Customer-Settings Package” is unique to each customer build based on customers specific configuration requirements. The final package called “Main Installation Package” installs TOE’s binaries and executables.

### Configuration

Most security related setting of the application configured by running Customer-Settings package setup. These settings can not be altered by the users once installed. Certain settings could be set to be completely invisible to users. The algorithms used for encryption and key sizes are all pre-determined and hard-coded. The password for the certificate store is set by the user using Kermen DEPO.

Kermen SUR can be used to encrypt/decrypt files once file operation settings configured such as where the encrypted files will be copied, to whom the files will be encrypted, by whom the encrypted files will be signed .

Any number of folders can be set by KERMEN GDM where the content is automatically encrypted to selected users. Encryption can also be done based on user Access rights on specified folders.

KERMEN MGM and EEPM use LDAP such as Active Directory to access certificates and public keys of the email recipients in order to encypt and sign email and file content. KERMEN MGM can also utilize local certificate store KERMEN DEPO.

### TOE Operation and Use

#### **Intended Use**

By using KERMEN, two or more individuals can exchange data securely over unprotected communication paths, eg. Networks, without risking any unauthorized persons reading the confidential data. To achieve this, the data is encrypted before it is transferred between the two parties, and thus made unreadable for anyone who are not included in the recipients of the data defined by the sender.

The file encryption/decryption can be achieved using either of KERMEN SUR, KERMEN MGM and KERMEN GDM applications.

Kermen SUR can perform file encryption/decryption in just one step just by dragging the files to be encrypted/decrypted onto the application window.

KERMEN MGM Shell extension enables individual encryption/decryption & signing operations on files. Any encypted file when double clicked automatically decrypts via KERMEN MGM.

KERMEN EEPM signs and/or encrypts email messages to recipient(s) so that only the recipient(s) with private key can decrypt.

If the user’s private key stored in a smartcard then the smartcard must be plugged in for cryptographic operations. For the keys stored in the certificate store, store password must be provided by the user.

#### **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.

#### **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 wrapped for the recipients using associated asymmetric public keys. AES symmetric encryption algorithm is used in GCM and CBC mode 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 ensure integrity of data files, encrypted files are signed by the sender’s private key before the encryption done. CMS format is used to sign and create SignedData as described in [RFC5652].Thesignaturealgorithm defined in the sender’s public key is used as the signature algorithm.
* **Encryption of emails:** Emails are encrypted using the S/MIME standard defined in [RFC 5751].
* **Integrity protection of emails:** To ensure integrity of emails, encrypted emails are signed using the S/MIME standard defined in [RFC 5751].
* **Decryption of data:** Encrypted data is decrypted if the user posseses the corresponding private key in his smart card or certificate store. Only the recipients of the email set by sender can decrypt data.
* **Integrity Check of data:** Encrypteddatais also signed. After decrpytion, the output signed data can be validated to check integrity of the original data.
* **Nonrepudiation of data:** The private keys used in signing data are associated with a X.509 certificate points to the identity of the users which are undeniable. By the nature of the signature operation, user can’t claim that the data is not signed by himself or modified by someone else.
* **Key Storage:** Soft private keys are stored in the certificate store and protected by a password. The algorithm PBKDF2 desribed in [RFC 8018][[5]](#_A.2_References) (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 store password must be provided.

### TOE Environment and Physical Protection

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

## Related Standards and Documents

|  |  |
| --- | --- |
| [CC] | Common Criteria (CC) for Information Technology Security  Evaluation, Version 3.1, Revision 5, April 2017.  » Part 1: Introduction and general model. April 2017.  Version 3.1. CCMB-2017-04-001  » Part 2: Security Functional Components. April 2017.  Version 3.1. CCMB-2017-04-002  » Part 3: Security Assurance Requirements. April 2017.  Version 3.1. CCMB-2017-04-003 |
| [CEM] | Common Methodology for Information Technology Security  Evaluation. Evaluation Methodology. April 2017. Version 3.1, Revision 5,  CCMB-2017-04-004 |
| [GPPS] | ISO/IEC TR 15446:2009, Guide for the production of  Protection Profiles and Security Targets, Second edition 2009-03-01 |
| NIST SP 800-88 | Guidelines for Media Sanitization, Revision 1, December 2014 |
| [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: 5652, 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 |
| [RFC 5751] | Request For Comments: 5751, Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification |
| [RFC 6090] | Fundamental Elliptic Curve Cryptography Algorithms, February 2011 |
| 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 |

# Conformance Claim

## CC Conformance Claim

This ST claims conformance to

* Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2017-04-001, Version 3.1, Revision 5, April 2017
* Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2017-04-002, Version 3.1, Revision 5, April 2017, CC Part 2, Conformant
* Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; CCMB-2017-04-003, Version 3.1, Revision 5, April 2017, CC Part 3, Conformant

The

* Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; CCMB-2017-04-004, Version 3.1, Revision 5, April 2017

has to be taken into account.

## PP Claim

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

## Package Claim

EAL4+ (ALC\_FLR.2)

## Conformance Rationale

The assurance level selected for this type of TOE is EAL4 +(ALC\_FLR.2). This evaluation assurance package was selected to permit a developer to gain maximum assurance from positive security engineering based on good commercial development practices which, though rigorous, don’t require substantial specialist knowledge, skills, and other resources.

Augmentation results from the selection of ALC\_FLR.2 Flaw Reporting Procedures.

In order to provide a meaningful level of assurance and adequate level of defense against such attacks that the TOE provides, 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). Since the TOE is security related, the tracking of security flaws is a very reasonable expectation and within the bounds of standard, best commercial practice. EAL 4 augmented is deemed appropriate to satisfy customers’ expectations for trusted certificate authorities.

# SecurIty Problem DEFINITION

## Secure Usage Assumptions

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

|  |  |
| --- | --- |
| **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 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 user 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 according to KERMEN User Manual document [[6]](#_A.2_References) |
| **A.CONNECT** | The single user PC which the TOE is running is not connected directly to any 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 Mail and LDAP servers resides. That is also protected against attacks. |

## Organizational Security Policies

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

## Threats

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

### 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  (primary asset) | Any file or email that contains information to  be protected. | User Data |
| Smartcard PIN | The PIN for the smartcard login | User Data |
| Certificate Store Password | The password for the certificate store | User Data |
| Private Keys | Asymmetric private keys used for data 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 data is transferred, e.g., networks or other paths of transmission where communication media like CDs, DVDs including the encrypted data could be shared. |

Table 2: Agents

### 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 data is transferred succeeds in accessing the content of the data, i.e. the attacker violates the confidentiality of the information included in the data.  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 as such has to perform  cryptanalysis to reveal the underlying plain text of the encrypted data. |
| Asset | Data |
| Agent | Attacker |

|  |  |
| --- | --- |
| **Threat:** T.MODIFY – Loss of integrity | |
| Attack | An attacker of one of the communication paths over which the KERMEN data is transferred modifies the data, i.e. replacing or changing the content of the data in a way that is not detected.  The attack is achieved by interrupting the transfer due to possess the data to accomplish an active attack violating the integrity of the information included in the data before sending it to the receiver. Therefore, the attacker has either to break the integrity protection of the data, modifying the content of the data and reconstructing the protection again. Or the attacker replaces the whole data and constructs the integrity protection. Afterwards the data is sent to the intended destination. In both cases the attacker must either possess the right key used for integrity protection or perform cryptanalysis to reveal the right key. |
| Asset | Data, Private Keys |
| 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 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 data 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 necessary to successfully gain the plain data 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. Yet an attacker will easily come to the point that the attacks described in T.DISCLOSE and T. MODIFY has to be combined to be successful in violating the integrity and/or confidentiality of the data content since the transferred data is signed and encrypted.

Attacks which modify the content of the transferred data without breaking the integrity protection are as well conceivable. An attacker may completely intercept the communication so that data does not reach the intendent destination. This attacks have the same effects as errors occurring during communication. Primarily, the availability of the information transferred is violated. The receiver fails e.g. in validating the integrity of the data, the data will not be decrypted. This attack 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 their 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 data is in general available for an attacker.

# 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.

## 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 data 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 data ( i.e. replacing or modifying the content of the file); mechanisms must be provided to detect loss of integrity of the information in the data. |
| **O.ALGORITHM** | The TOE must only allow the use of approved encryption algorithms and key lengths, i.e. AES and 256 bit. |

## 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 data 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 any untrusted network. This means that the PC must either not be connected to any networks or it must be connected to a trusted network in which Mail and LDAP servers resides which is also protected against attacks. Therefore there are no undocumented security critical side effects on the security functions of the TOE are coming from this network. |

## Security Objectives Rationale

### 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 |  | 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

### 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 data 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 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 data 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. OE.PASSWORDDIS requires in addition that the TOE only uses 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.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 potential attackers in opportunities to decrypt the keystore and get access to the key used for validating the integrity of transmitted data.  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. T.MODIFY is also diminished by OE.PHYSICAL which requires attacker can not access target machine via unauthorized ways. |

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 in 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 the user 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. Since EEPM is only used intranet, this is also valid for EEPM. Therefore there are 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 in which Mail and LDAP servers resides too that is protected against attacks. Therefore, there are 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 that 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

# Extended components defınıtıons

This ST does not define extended components

# IT Securıty 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.6 | Re-authenticating |
| FIA\_UAU.7 | Protected authentication feedback |
| FMT\_MSA.1 | Management of security attributes |
| FMT\_MSA.3 | Static attribute initialization |
| FMT\_SMF.1 | Specification of Management Functions |

Table 8 Functional Requirements on the TOE

The TOE implements two Security Function Policy (SFP) called private key management access control SFP and public key information flow control SFP.

The first one, private key management control SFP, indicates that it is a policy regulating how to access and manage private keys. The SFP regulates that when a user wants to access a private key in the certificate store or smartcards must provide the corresponding password/PIN.

To be able to export a private key, a user first needs to provide the password of the certificate store and then the password of that private key.

Both passwords of the private key and the sertificate store must be provided to be able to import a private key into the sertificate store .

Current password of the certificate store must be provided to change it with a new password.

The second SFP called public key information flow control SFP, indicates that it is a policy regulating how to access, retrieve and manage public key Certificates and associated public keys stored in LDAP servers. TOE always controls public keys validity via OCSP or CRL (Certificate Revocation Lists) it works with.

## 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 are used in this ST.

The **refinement** operation is used to add details 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.

## TOE Security Functional Requirements

### Class FCS - Cryptographic Support

#### 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: AES],** and specified cryptographic key sizes**, [assignment: 256 bits],** that meet the following: **[assignment conform to the FIPS 197].**  **Application Note:** Cryptographic keys generated randomly by the Random Number Generator. |

#### 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 symmetric keys and private keys.]** that meet the following:**[assignment: none]**  **Application Note:** Zeroization of memory allocated to symmetric keys used in encryption and decryption of data and the private keys just after the data operation is completed. Symmetric keys and private keys are not stored in anywhere else |  |

#### FCS\_\_COP.1/KEYENC/RSA – 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: default value is 2048 bits, bigger than 2048 bits also supported]** that meet the following: **[assignment: conform to the FIPS 140-2 Annex.A]**  **Application Note:** cryptographic key size of less than 2048 bits is also supported by TOE but it is not counted to be secure. Therefore, key size less than 2048 bits is out of scope for evaluation of TOE. |

#### FCS\_\_COP.1/KEYENC/ Elliptic Curve – 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: Elliptic Curve]** and cryptographic key sizes **[assignment: 384 bits]** that meet the following: **[assignment: conform to the FIPS 140-2 Annex.A]** |

#### FCS\_\_COP.1/KEYDEC/ RSA – 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: default value is 2048, bigger than 2048 also supported for RSA]** that meet the following: **[assignment: conform to the FIPS 140-2 Annex.A]**  **Application Note:** cryptographic key size less than 2048 bits is also supported by TOE but it is not considered secure. Therefore, key size less than 2048 bits is out of scope for evaluation of TOE. |

#### FCS\_\_COP.1/KEYDEC/Elliptic Curve – 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: Elliptic Curve]** and cryptographic key sizes **[assignment: 384 bits]** that meet the following: **[assignment: conform to the FIPS 140-2 Annex.A]** |

#### FCS\_\_COP.1/ENC – Cryptographic Operation (data encryption)

|  |  |
| --- | --- |
| *FCS\_COP.1.1* | The TSF shall perform **[assignment: encryption of data]** in accordance with a specified cryptographic algorithm, **[assignment: AES]** and cryptographic key sizes **[assignment: 256 bits]** that meet the following: **[assignment: conform to the FIPS 197]** |

#### FCS\_\_COP.1/DEC – Cryptographic Operation (data decryption)

|  |  |
| --- | --- |
| *FCS\_COP.1.1* | The TSF shall perform **[assignment: decryption of encrypted data]** in accordance with a specified cryptographic algorithm **[assignment: AES]** and cryptographic key sizes **[assignment: 256 bits]** that meet the following:**[assignment: conform to the FIPS 197].** |

#### FCS\_\_COP.1/SIGN /RSA – Cryptographic Operation (data signing)

|  |  |
| --- | --- |
| *FCS\_COP.1.1* | The TSF shall perform **[assignment: signingdata]** in accordance with a specified cryptographic algorithm, **[assignment: RSA]** and cryptographic key sizes **[assignment: default value is 2048 bits, bigger than 2048 bits also supported]** that meet the following: **[assignment: conform to the RFC 4056 and RFC 5751]**.  **Application Note:** Cryptographic key size less than 2048 bits is also supported by TOE but it is not considered secure. Therefore, key size less than 2048 bits is out of scope for evaluation of TOE. |

#### FCS\_\_COP.1/SIGN/Elliptic Curve – Cryptographic Operation (data signing)

|  |  |
| --- | --- |
| *FCS\_COP.1.1* | The TSF shall perform **[assignment: signingdata]** in accordance with a specified cryptographic algorithm, **[assignment: Elliptic Curve]** and cryptographic key sizes **[assignment: 384 bits]** that meet the following: **[assignment: conform to the RFC 6090 and RFC 5751]**. |

#### FCS\_\_COP.1/VERIFY/RSA – Cryptographic Operation (data verification)

|  |  |
| --- | --- |
| *FCS\_COP.1.1* | The TSF shall perform **[assignment: verification of signeddata]** in accordance with a specified cryptographic algorithm [**assignment: RSA]** and cryptographic key sizes **[assignment: default value is 2048 bits, bigger than 2048 bits also supported]** that meet the following: **[assignment: conform to the RFC 4056 and RFC 5751]**.  **Application Note:** cryptographic key size less than 2048 bits is also supported by TOE but it is not considered secure. Therefore, key size less than 2048 bits is out of scope for evaluation of TOE. |

### Class FDP – User Data Protection

#### FDP\_ACC.1 – Subset access control (Private Key Management Control)

|  |  |
| --- | --- |
| *FDP\_ACC.1.1* | The TSF shall enforce **[assignment: the private key management access control SFP]** on **[assignment: all users, The private keys in the certificate store and in the smartcards]** |

#### FDP\_ACF.1 – Security attribute based access control

|  |  |
| --- | --- |
| *FDP\_ACF.1.1* | The TSF shall enforce **[assigment: the private key management access control SFP]** to objects based on the following: **[assignment: user, private keys 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**] |

#### FCS\_\_COP.1/VERIFY/ Elliptic Curve – Cryptographic Operation (data verification)

|  |  |
| --- | --- |
| *FCS\_COP.1.1* | The TSF shall perform **[assignment: verification of signeddata]** in accordance with a specified cryptographic algorithm **[assignment: Elliptic Curve]** and cryptographic key sizes **[384 bits]** that meet the following: **[assignment: conform to the RFC 6090 and RFC 5751]**. |

#### FDP\_ITC.1 - Import of user data without security attributes

|  |  |
| --- | --- |
| *FDP\_ITC.1.1* | The TSF shall enforce the **[assignment:** **the private key management access control SFP and public key information flow 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]**. |

#### FDP\_ETC.1 Export of user data without security attributes

|  |  |
| --- | --- |
| *FDP\_ETC.1.1* | The TSF shall enforce the **[assignment: the private key management access control SFP and public key information flow 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. |

#### 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 or signed/ encrypted data]** |
| *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. |
|  |  |

### Class FIA – Identification and Authentication

#### 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. |

#### 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. |

#### FIA\_UAU.6 – Re-authenticating

|  |  |
| --- | --- |
| *FIA\_UAU.6.1* | The TSF shall re-authenticate the user under the conditions **[assignment: 60 minutes later after the user provide a corresponding password for the certificate store]** |

#### 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. |

### Class FMT – Security Management

#### FMT\_MSA.1 – Management of security attributes

|  |  |
| --- | --- |
| *FMT\_MSA.1.1* | The TSF shall enforce the **[assignment:** **private key management 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 certificate store and smartcard access via password. The TOE is assumed to be operated on a single user machine with only one user having access to the TOE.

#### FMT\_MSA.3/PRIVATE – Static attribute initialization

|  |  |
| --- | --- |
| *FMT\_MSA.3.1* | The TSF shall enforce the **[assignment:** **private key management access control SFP]** to provide **[selection:** **restrictive]** default values for security attributes that are used to enforce the SFP. |
| *FMT\_MSA.3.2* | The TSF shall allow **[assignment: no user]** 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 certificate store passwords; when the application runs for the first time, a password has to be set.

#### FMT\_MSA.3/PUBLIC – Static attribute initialization

|  |  |
| --- | --- |
| *FMT\_MSA.3.1* | The TSF shall enforce the **[assignment:** **public key information flow control]** to provide **[selection:** **permissive]** default values for security attributes that are used to enforce the SFP. |
| *FMT\_MSA.3.2* | The TSF shall allow **[assignment: no user]** 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 certificate store passwords; when the application runs for the first time, a password has to be set.

#### 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 and public keys (only soft keys)** * **import private and public keys (only soft keys)** * **delete private and public keys (only soft keys) ]** |

**Application Note.1:** Import requires both the private key and certificate store password to be provided.

**Application Note.2:** Export requires both the certificate store and private key passwords to be provided.

**Application Note.3:** Password change of certificate store requires current certificate store password to be provided.

## 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

## Security Requirements Rationale

### 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 | X |  |
| FCS\_COP.1/KEYENC/RSA |  | X | X |
| FCS\_COP.1/KEYENC/ Elliptic Curve |  | X | X |
| FCS\_COP.1/KEYDEC/ RSA |  | X | X |
| FCS\_COP.1/KEYDEC/ Elliptic Curve |  | X | X |
| FCS\_COP.1/ENC |  | X | X |
| FCS\_COP.1/DEC |  | X | X |
| FCS\_COP.1/SIGN/RSA | X |  | X |
| FCS\_COP.1/SIGN/ Elliptic Curve | X |  | X |
| FCS\_COP.1/VERIFY/RSA | X |  | X |
| FCS\_COP.1/VERIFY/ Elliptic Curve | X |  | X |
| FDP\_ACC.1 | X | X |  |
| FDP\_ACF.1 | X | X |  |
| FDP\_DAU.2 | X |  | X |
| FDP\_ETC.1 | X | X |  |
| FDP\_ITC.1 | X | X |  |
| FIA\_UID.1 | X | X |  |
| FIA\_UAU.1 | X | X |  |
| FIA\_UAU.6 | X |  |  |
| FIA\_UAU.7 | X |  |  |
| FMT\_MSA.1 | X | X |  |
| FMT\_MSA.3/PRIVATE | X | X |  |
| FMT\_MSA.3/PUBLIC | X | X |  |
| FMT\_SMF.1 | X | X |  |

Table 10 TOE Security objectives meeting SFRs

### 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 data is achieved by the cryptographic operations FCS\_COP.1/SIGN/RSA-FCS\_COP.1/SIGN/Elliptic Curve together with FCS\_COP.1/VERIFY/RSA- FCS\_COP.1/VERIFY/Elliptic Curve. * The data is signed by the sender FCS\_COP.1/SIGN/RSA-FCS\_COP.1/SIGN/ Elliptic Curve so, the receiver can verify and detect any integrity violations FCS\_COP.1/VERIFY/Elliptic Curve-FCS\_COP.1/VERIFY/RSA. * FCS\_CKM.4 ensures that private keys used been destructed properly right after cryptographic operations. * FDP\_ACC.1 and FDP\_ACF.1 ensures that only authorized users can manage and access private keys. * 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 certificate store password can change it. * With FMT\_MSA.3, only users who know the initial password of the soft keys can manage the private keys. * FMT\_SMF.1 provides the specific management functions for private and public key import/export. It also provides a function for changing certificate store password. * FIA\_UID.1 and FIA\_UAU.1 ensures that only authenticated users have access to signing private keys and authentication feedback is protected. |
| O.DISCLOSE | * The mechanism to protect data during transmission against violation of confidentiality is achieved by the cryptographic operation FCS\_COP.1/KEYENC/RSA-FCS\_COP.1/KEYENC/ Elliptic Curve and FCS\_COP.1/ENC ensuring the data encrypted in such a way, confidentiality of the content is protected and only the recipients who have associated private keys can decrypt. * The mechanism to provide data decryption is achieved by the cryptographic operation FCS\_COP.1/DEC and FCS\_COP.1/KEYDEC/RSA- FCS\_COP.1/KEYDEC/Elliptic Curve.Data decryption on the receivers side is strictly speaking not needed to achieve integrity protection but impelemented to provide access to the encrypted information to those who are authorized. * FCS\_CKM.1 ensures generation of the symmetric encryption keys. Encryption mechanisms achieves that only cryptographic keys generated first therefore supporting requirements regarding these keys are needed. Symmetric encryption keys are generated on the sender side wrapped with asymmetric public key, to be decrypted by associated private key on the receiver side. * FCS\_CKM.4 ensures that all symmetric keys and private keys stored in certificate store used for content encryption is destructed right after use. * With FMT\_MSA.1 only users who know the certificate store password can change it. * With FMT\_MSA.3 only users who know the initial password value of the soft keys can manage the private keys. * FMT\_SMF.1 provides functions to import/export private and public keys and change certificate store password. * FIA\_UID.1 and FIA\_UAU.1 ensures that only authenticated user has access to key-encryption private keys. * FIA\_UAU.7 and FIA\_UAU.6 ensures that authentication feedback is protected. * FDP\_ETC.1 and FDP\_ITC.1 ensures that the user private and public keys are imported into and exported from the certificate store with SFP. |
| 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. * Only approved algorithms with specified key sizes are allowed for cryptographic operations in the cryptographic operation requirements   + FCS\_COP.1/KEYENC/RSA-FCS\_COP.1/KEYENC/Elliptic Curve,   + FCS\_COP.1/KEYDEC/RSA-FCS\_COP.1/KEYDEC/Elliptic Curve,   + FCS\_COP.1/ENC,   + FCS\_COP.1/DEC,   + FCS\_COP.1/SIGN/RSA-FCS\_COP.1/SIGN/Elliptic Curve,   + FCS\_COP.1/VERIFY/RSA-FCS\_COP.1/VERIFY/Elliptic Curve.   These algorithms are fixed and can’t be managed by the user (see FMT\_SMF.1 – no management function exists).   * FDP\_DAU.2.1 and FDP\_DAU.2.2 ensures that transferred data is digitally signed and verified upon receival. |

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.

### Rationale of Selected Assurance Level

The assurance level EAL4+ (ALC\_FLR.2) has been chosen as the most appropriate level as it provides sufficient level of assurance for an application that is encrypting data in a secure and well managed environment. The attacker is also assumed only to attack exported or imported data from TOE and not the TOE itself, thereby limiting the opportunity of an attacker.

### 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.

#### 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/RSA | [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 FDP\_ITC.1  The lack of FCS\_CKM.4 is justified below this table. |
| FCS\_COP.1/KEYENC/ Elliptic Curve | [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 FDP\_ITC.1  The lack of FCS\_CKM.4 is justified below this table. |
| FCS\_COP.1/KEYDEC/RSA | [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.4  Fulfilled by FDP\_ITC.1 |
| FCS\_COP.1/KEYDEC / Elliptic Curve | [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.4  Fulfilled by FDP\_ITC.1 |
| 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 FDP\_ITC.1,  Fulfilled by FCS\_CKM.4 |
| FCS\_COP.1/SIGN/RSA | [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 FDP\_ITC.1,  Fulfilled by FCS\_CKM.4 |
| FCS\_COP.1/SIGN / Elliptic Curve | [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 FDP\_ITC.1,  Fulfilled by FCS\_CKM.4 |
| FCS\_COP.1/VERIFY /RSA | [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 FDP\_ITC.1,  The lack of FCS\_CKM.4 is justified below this table |
| FCS\_COP.1/VERIFY / Elliptic Curve | [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 FDP\_ITC.1,  The lack of FCS\_CKM.4 is justified below this table |
| FDP\_ACC.1 | FDP\_ACF.1 Security attribute based acc ess 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 | Fulfilled 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 initialization | 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.6 | No dependencies | n.a. |
| FIA\_UAU.7 | FIA\_UAU.1 Timing of authentication | Fulfilled by FIA\_UAU.1 |
| 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.3/PRIVATE | 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\_MSA.3/PUBLIC | 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.

***Justification of lack of FCS\_CKM.4:***

Public keys are not destructed after they use for cryptographic operations.

#### 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.

### Internal Consistency and Mutual Support of SFRs

Section 6.4.2 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/e-mails 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). First seed value needs to be securely generated using a hardware RNG (random number generator) such as smartcards. If smartcards are not applicable, seed value needs to be generated by software using the operating system resources as described in 7.1.1

Provided users have private key in the certificate store, store password has to be provided once, before TOE can decrypt any encrypted data. The keys are derived by PKCS12 PBKDF2 key derivation algorithm. Users can perform certain management functions on the certificate store, such as importing new private keys or exporting existing keys only if they can provide correct password. (FMT\_SMF.1.) The certificate store password can also be changed. When private keys are stored in smartcards, the security of the keys are provided by the smartcards. Private keys in smartcard are only accessible after the PIN is provided. It is impossible to extract private keys from the smartcards.

User A can encrypt data for user B by using User B’s public key placed in User B’s certificate. Before the encryption, plain data is signed by the user’s signing private key placed in the smartcard or the certificate store, by providing the required PIN/password. The signature protects the data against integrity attacks and corruptions. After signed and encrypted data envelope is created, it can be transported in insecure environments. The recipient of the encrypted data can decrypt it by using his own private key as explained previously. Then user verifies the signature of the sender and accesses the plain data.

# TOE Summary Specıfıcatıon

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.

## 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.

### SF.KEYGEN - Key Generation

Symmetric encryption keys are created randomly by using the seed provided by smartcards. If smartcards are not applicable seed generated by software using the operating system resources. Software seed generator uses OpenSSL Random number generator module. OpenSSL RNG module conformed to NIST SP800-90A [7].

### 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/Elliptic Curve and symmetric encryption of the data with AES algorithm in GCM and CBC mode with a 256-bits key. The format of the encrypted data in both cases is based on EnvelopedData format [RFC 5652].

### SF. MAIL\_CRYPT - Mail encryption/decryption

Mails are encrypted and decrypted using the S/MIME standard including asymmetric encryption of symmetric keys with RSA/Elliptic Curve and content encryption of the mail with AES algorithm in GCM and CBC mode with a 256-bits key [RFC 5751].

### 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/Elliptic Curve 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].

### SF.MAIL\_SIGN – Mail signing

Mails are signed and verified using the S/MIME standard (Signed Data) including asymmetric signature of data with RSA / Elliptic Curve, applying a signature to a message provides message integrity, authentication and non-repudiation of origin [RFC 5751].

### SF.CHECK\_FILE\_INTEGRITY – File verification

The first step of the validation is that make sure the file is in CMS SignedData format and the integrity check is implemented as stated in RFC 5652. Only if SF.CHECK\_FILE\_INTEGRITY is successful, decryption according to SF.FILE\_CRYPT is performed with the associated private key.

### SF.CHECK\_MAIL\_INTEGRITY – Mail verification

Digitally signed mail provides the verification of the data, without first verifying that the message actually signed and knowing the state of the signature, this can lead to incorrect handling of the message. [RFC 5751]

The first step of the validation is to make sure the mail is in CMS SignedData format and the integrity check is implemented as stated in RFC 5652. Only if SF.CHECK\_MAIL\_INTEGRITY is successful, decryption according to SF.MAIL\_CRYPT is performed with the corresponding private key.

### SF.MANAGE

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

* delete private and public keys (only soft keys)
* change certificate store password
* export private and public keys (only soft keys)
* import private and public keys (only soft keys)

Export private key function requires the user both to present the password of the certificate store and the password of the private key.

Import private key function requires the user both to present the password of the private key and the password of the certificate store.

Change certificate store password function requires the user to present the correct password of the certificate store.

Key unwrap password must be assigned by the user to export private keys. Key unwrap password of the imported soft keys must be entered by the user to import private keys. Current password has to be provided to change the certificate password.

### SF.CLEAR

Encryption keys and soft private keys (conform to thePKCS12 standards) are cleared from the memory just after their use and the memory locations allocated are zeroized.

## 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 data 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.MAIL\_CRYPT | SF.MAIL\_SIGN | SF.CHECK\_FILE\_INTEGRITY | SF.CHECK\_MAIL\_INTEGRITY | SF.MANAGE | SF.CLEAR |
| FCS\_CKM.1 | X |  |  |  |  |  |  |  |  |
| FCS\_CKM.4 |  |  |  |  |  |  |  |  | X |
| FCS\_COP.1/KEYENC/RSA |  | X |  | X |  |  |  |  |  |
| FCS\_COP.1/KEYENC/Elliptic Curve |  | X |  | X |  |  |  |  |  |
| FCS\_COP.1/KEYDEC/ RSA |  | X |  | X |  |  |  |  |  |
| FCS\_COP.1/KEYDEC/Elliptic Curve |  | X |  | X |  |  |  |  |  |
| FCS\_COP.1/ENC |  | X |  | X |  |  |  |  |  |
| FCS\_COP.1/DEC |  | X |  | X |  |  |  |  |  |
| FCS\_COP.1/SIGN/RSA |  |  | X |  | X |  |  |  |  |
| FCS\_COP.1/SIGN/ Elliptic Curve |  |  | X |  | X |  |  |  |  |
| FCS\_COP.1/VERIFY/RSA |  |  |  |  |  | X | X |  |  |
| FCS\_COP.1/VERIFY/ Elliptic Curve |  |  |  |  |  | X | X |  |  |
| FDP\_ACC.1 |  | X | X | X | X |  |  | X |  |
| FDP\_ACF.1 |  | X | X | X | X |  |  | X |  |
| FDP\_DAU.2 |  |  | X |  | X | X | X |  |  |
| FDP\_ETC.1 |  |  |  |  |  |  |  | X |  |
| FDP\_ITC.1 |  |  |  |  |  |  |  | X |  |
| FIA\_UID.1 |  |  | X |  | X |  |  | X |  |
| FIA\_UAU.1 |  |  | X |  | X |  |  | X |  |
| FIA\_UAU.6 |  |  | X |  | X |  |  | X |  |
| FIA\_UAU.7 |  |  | X |  | X |  |  | X |  |
| FMT\_MSA.1 |  |  |  |  |  |  |  | X |  |
| FMT\_MSA.3/PRIVATE |  |  |  |  |  |  |  | X |  |
| FMT\_MSA.3/PUBLIC |  |  |  |  |  |  |  | X |  |
| FMT\_SMF.1 |  |  |  |  |  |  |  | X |  |

Table 13 TOE Security Functions meeting SFRs and Vice Versa

### 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 destruction 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 data encryption is satisfied by the security function SF.FILE\_CRYPT and SF.MAIL\_CRYPT, which specifies that asymmetric encryption algorithm specified in user certificates is used for the encryption of symmetric keys and AES in GCM and CBC mode with a 256 bit is used for data encryption, conforming to [RFC 5652 and RFC 5751 for mail encryption]. The requirement for the data signing is satisfied by the security function SF.FILE\_CHECK\_INTEGRITY, SF.CHECK\_MAIL\_INTEGRITY, SF\_MAIL\_SIGN and SF\_FILE\_SIGN which specifies the hash and signature algorithms are used in the data signing conforming to [RFC 5652 and RFC 5751 for mail signing]. |
| 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, SF.MAIL\_CRYPT, SF.MAIL\_SIGN 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, SF.MAIL\_CRYPT, SF.MAIL\_SIGN and SF.FILE\_SIGN security functions implementing that users must present the correct certificate store password before accessing the private keys. |
| FDP\_DAU.2 | The requirement for data authentication with identity of guarantor is satisfied by the security function SF.CHECK\_FILE\_INTEGRITY, SF\_CHECK\_MAIL \_INTEGRITY, SF\_FILE\_SIGN and SF\_MAIL\_SIGN which implements signing and verification of encrypted data |
| 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 and public 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 and public keys into the certificate store. |
| FIA\_UID.1 | The requirement for timing of identification is satisfied by the security function SF.MANAGE, SF.MAIL\_SIGN 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.MANAGE, SF.MAIL\_SIGN and SF\_FILE\_SIGN implementing the private key access control SFP for all users accessing the private key. |
| FIA\_UAU.6 | The requirement for re-authentication is satisfied by the security function SF.MANAGE, SF.MAIL\_SIGN and SF\_FILE\_SIGN After sixty minutes cache for password expired and user must reenter the password for authentication. |
| FIA\_UAU.7 | The requirement for protection of authentication is satisfied the security function, SF.MANAGE, SF.MAIL\_SIGN 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.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. |

### 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. 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.

### 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

# Appendıx

## 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 |
| **S/MIME** | Secure Multipurpose Internet Mail Extension |
| **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 |

## A.2 References

[1] RFC 5652 Cyrptographic Message Syntax (CMS)

[2] RFC 5751 Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification

[3] RFC 5280 Internet X-509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

[4] NISP SP 800-88 Rev.1 Guidelines for Media Sanitization

[5] RFC 8018 PKCS #5 Password-Based Cryptography Specification Version 2.1

[6] KERMEN Kullanım Kılavuzu, Bilişim ve Bigi Güvenliği İleri Teknolojiler Araştırma Merkezi (BİLGEM)/ TÜBİTAK, Sürüm 1.3, 25 Eylül 2019

[7] NIST SP 800-90A Rev.1 Recommendation for Random Number Generation Using Deterministic Random Bit Generators