TÜBİTAK UEKAE

ULUSAL ELEKTRONİK & KRİPTOLOJİ ARAŞTIRMA ENSTİTÜSÜ

ASGS PROJESİ AKILLIKART OKUYUCU PROJE GRUBU

KKEC APPLICATION SOFTWARE VERSION 1.41.06A

SECURITY TARGET LITE

Revision No	1.0
Revision Date	07.12.2011
Document Code	KKEC_UY_ST_LITE
File Name	KKEC_UY_ST_LITE
Prepared by	
Mustafa SELVİ	Chief Researcher
Approved by	
Ercan ÖLÇER	Project Manager

Revision History

Revision No	**Revision Reason**	**Date of Revision**
1.0	First Publication.	08.04.2009

CONTENT

ABBREVIATIONS

3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
AKİS	Akıllı Kart İşletim Sistemi (Smartcard Operating System)
APDU	Application Protocol Data Unit
ASGS	Akıllı Kart Tabanlı Sosyal Güvenlik Sistemi (Smartcard Based Social Security System)
CC	Common Criteria
CCID	Chip/Smart Card Interface Devices
CPU	Central Processing Unit
CTN	Cihaz Takip Numarası (Device Track Number)
DC	Direct Current
DES	Data Encryption Standard
EAL	Evaluation Assurance Level
EDH	Ephemeral Diffie-Hellman
EKK	Elektronik Kimlik Kartı (Electronic Identity Card)
EKDS	Elektronik Kimlik Doğrulama Sistemi (Electronic Identity Verification System)
EU	Eczane Uygulaması (Pharmacy Application)
GEM	Güvenli Erişim Modülü (Security Access Module)
GSP	Güvenlik Servisleri Platformu (Security Services Platform)
HMAC	Hash Message Authentication Code
IC	Integrated Circuit
IK	Imza Kartı (Signature Card)
KDB	Kimlik Doğrulama Bildirimi (Identity Verification Assertion)
KDP	Kimlik Doğrulama Politikası (Identity Verification Policy)
KDPS	Kimlik Doğrulama Politika Sunucusu (Identity Verification Policy Server)
KDS	Kimlik Doğrulama Sunucusu (Identity Verification Server)
KKEC	Kurumsal Kart Erişim Cihazı (Institutional Smartcard Access Device)
KECÖB	Kart Erişim Cihazı Özelleştirme Birimi (Smartcard Access Device Personalization Unit)
KSTB	Kart Sahibinin Tek Belirleyicisi (Card Holder Unique Identifier)
MEDULA	Online Certificate Status Protocol
OCSP	Online Certificate Status Protocol
OCSPS	Online Certificate Status Protocol Server
OYA	Otomasyon Yazılımı Arabirimi (Automation Software Interface)
PGS	Performans Gözlem Sunucusu (Performance Observation Server)
PIN	Personal Identification Number
RSA	Rivest – Shamir – Adleman (RSA Algorithm)
SC	Smartcard
SGK	Sosyal Güvenlik Kurumu (Social Security Association)
SFR	Security Functional Requirement
SPS	Software Publisher Server
SSL	Secure Socket Layer
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TPDU	Transmission Protocol Data Unit
TSF	TOE Security Function
TÜBİTAK	Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (Scientific and Technologic Research Association of Turkey)

UEKAE	Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü (National Research Institute of Electronics and Cryptology)
USB	Universal Serial Bus
USB-CCID	Universal Serial Bus – Chip/Smart Card Interface Devices
USB-DFU	Universal Serial Bus – Device Firmware Upgrade

1. ST Introduction

1.1 ST Reference

KKEC Application Software Version 1.41.06A Security Target Lite, revision 1.0, 07.12.2011

1.2 TOE Reference

KKEC Application Software Version 1.41.06A

1.3 TOE Overview

TOE is the application software of KKEC (Institutional Smartcard Access Device developed by TUBITAK-UEKAE). It performs smartcard based personal identity verification and digital signature operations for services given over electronic media. TOE has the following features:

- Cardholder authentication by using PIN and biometrics (either fingerprint data or fingervein data),
- Authentication of EKK/IKs and authentication of KKEC using GEM (Security Access Module, available as a SIM card):
  - Authentication of EKK/IK by using asymmetric authentication method,
  - Authentication of GEM by using symmetric authentication method,
- Symmetric and asymmetric encryption and decryption using 128-bit DES3, 256-bit AES and 2048-bit RSA algorithms,
- HMAC using 256-bit SHA algorithm,
- Digital sign and sign verification using 2048-bit RSA algorithm,
- Provable non-repudiation
- Secure communication by using TLS v1.0,
- Automatic, remote and secure upgrade

TOE is to be distributed and used with only KKEC and it will manage only smartcards with AKİS (version 1.2.1i and 1.2.1n) and UKİS (version 1.2.1) operating systems for security reasons.

1.3.1 Operational Environment Components

1.3.1.1 Hardware Environment

TOE runs as the application software of KKEC. Therefore, the hardware components of KKEC compose the hardware environment of TOE.

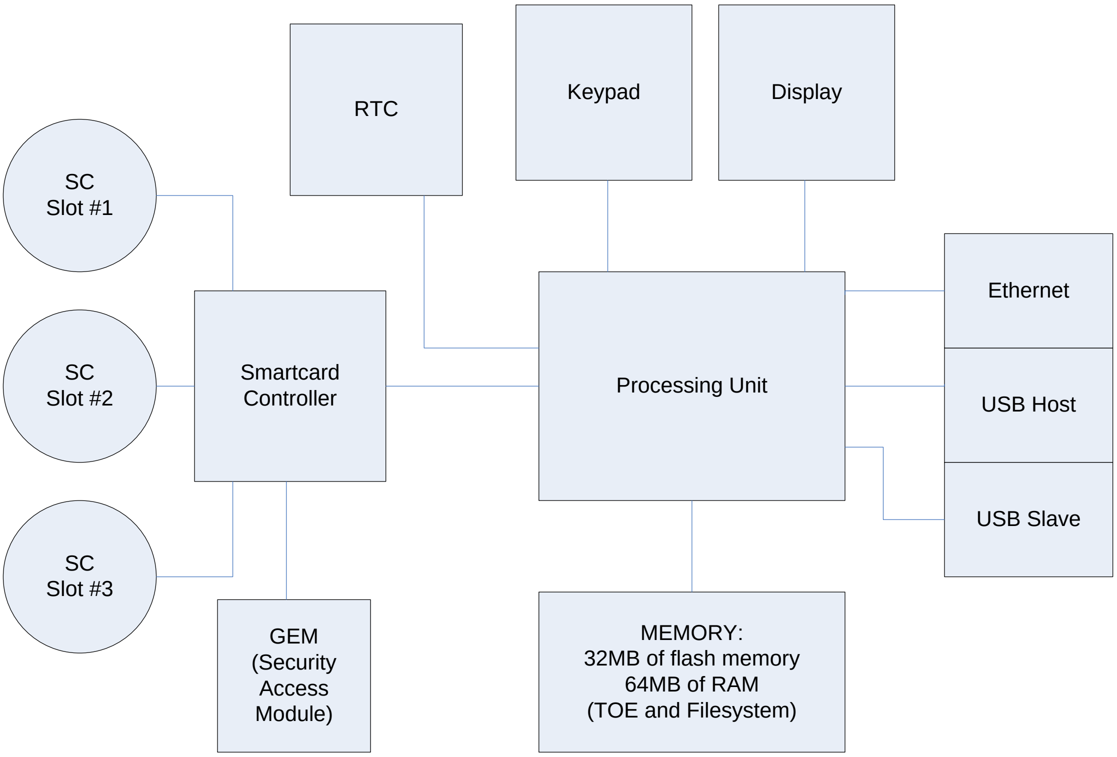

Figure 1. Hardware Environment Architecture of TOE

As shown in the block diagram in figure 1, KKEC includes:

- 200 MHz ARM920T core based processing unit,
- 32 MB of Flash Memory and 64 MB of SDRAM,
- Real Time Controller,
- 3 SC slots & 1 SIM card slot (compatible to IEC/ISO 7816),
- Security Access Module (GEM), placed into the SIM card slot
- 240x320 resolution TFT-LCD with 262K colors,
- 20-keys keypad,
- 128x128 pixels fingerprint sensor,
- USB 2.0 compliant full speed USB port for PC connection,
- 10 Mbit Ethernet port for network connection,
- USB 2.0 compliant full speed USB port for HUBC or external fingervein device connection,
- VGA port,
- +9V power supply input.

1.3.1.2 Software Environment

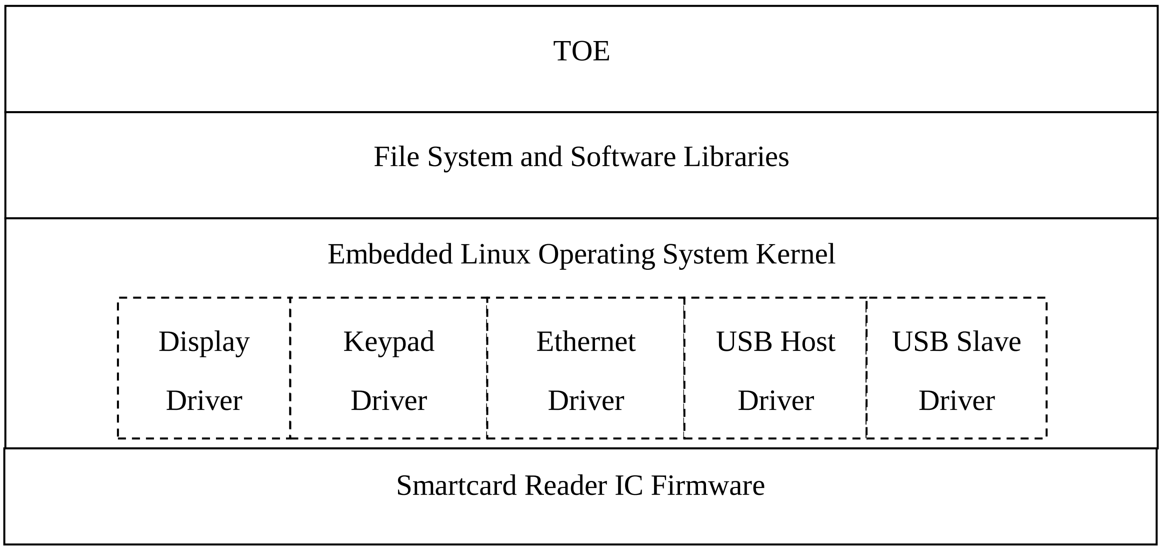
TOE
File System and Software Libraries
Embedded Linux Operating System Kernel
Display Driver
Keypad Driver
Ethernet Driver
USB Host Driver
USB Slave Driver
Smartcard Reader IC Firmware

Figure 2. Software Environment Architecture of TOE

TOE operates on an embedded linux environment (Kernel version 2.6.20.4) with a file-system in jffs2 format. The compiled kernel image (version: 01.02.07) comprises the OS kernel and some of the device drivers while the file-system (version: 01.10.06) is composed of the system files, the software libraries and the rest of the device drivers required by TOE. The file system also includes the TOE (KKEC Application Software Version 1.41.06A).

1.4 TOE Description

This part of the ST describes TOE with its general IT features as an aid to the understanding of its security requirements.

TOE is the application software which is loaded into the embedded flash memory of KKEC. It provides personal identity verification (PIV) and digital signature operations for smartcard based services over electronic media. TOE has the following features:

- Cardholder authentication by using PIN and biometrics (either fingerprint data or fingervein data),
- Authentication of EKK/IKs and KKEC using GEM (Security Access Module, available as a SIM card):
  - Authentication of EKK/IK by using asymmetric authentication method,
  - Authentication of GEM by using symmetric authentication method,
- Symmetric and asymmetric encryption and decryption using 128-bit DES3, 256-bit AES and 2048-bit RSA algorithms,
- HMAC using 256-bit SHA algorithm,
- Digital sign and sign verification using 2048-bit RSA algorithm,
- Provable non-repudiation,
- Secure communication by using TLS v1.0,
- Automatic, remote and secure upgrade

1.4.1 TOE User Environments

There are three user environments for TOE. These are hospital environment, pharmacy environment and family doctor office/unit environment. They are explained in the following subsections.

1.4.1.1 Hospital

In hospitals, as seen in figure 3 on the next page, the patient first goes to the reception office. Receptionist makes an examination record for the patient. As the first step of this process, identities of both the receptionist and the patient are verified by TOE with the help of their EKKs. Prior to authentication, KDP is determined by KDPS. If KDPS is unreachable for some reason, the policy is set as the default policy by TOE in which authentication with the highest security level (fourth level) is performed. During authentication, symmetric and asymmetric card verifications are performed and certificates of EKKs are validated online using OCSP to make sure the EKKs are valid and they are not revocated. According to KDP, if necessary, biometric identifications of users are also carried out by TOE using fingerprint or fingervein biometrics of the patient and the pharmacist. After authentication, the KDBs, which are electronically signed by TOE, are sent to GSP and then the GSP forwards them to KDS where they are registered. If all the processes are completed successfully, an examination record is created.

After the registration, the patient goes to the doctor. The doctor examines the patient and writes a prescription. To sign the prescription electronically, first, identity verification of the doctor and the patient are requested by GSP. After identity verification, KDBs prepared by TOE according to given KDP are sent to GSP. GSP is responsible to forward KDBs to KDS where they are registered.

Secondary, hospital automation software enables the doctor to write reports and prescriptions for the patient. After completed by the doctor, hospital automation software sends a sign request to GSP. Once received the sign request, GSP uses TOE to make the prescription signed electronically by both the doctor's IK and GEM to guaranty non-repudiation of operation by service requester or service participant and also to guaranty location of operation.

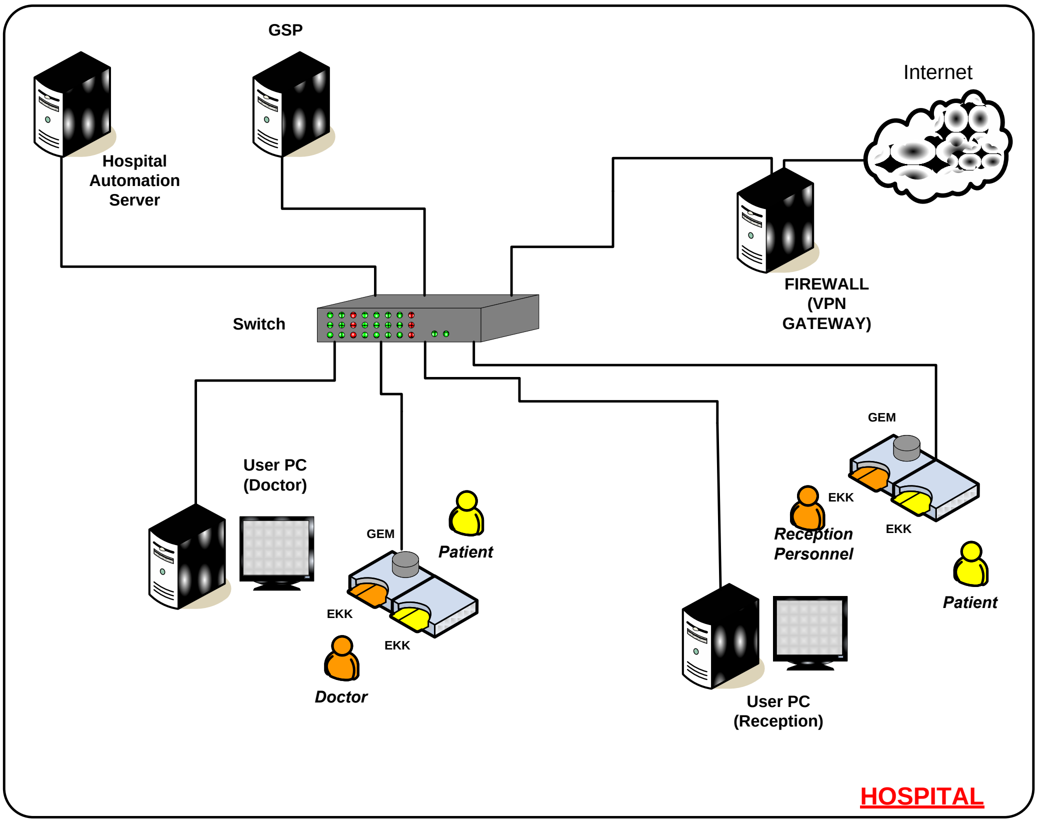

Figure 3. Hospital environment

1.4.1.2 Pharmacy

As seen in figure 4 on the next page, the patient goes to a pharmacy. The pharmacist or an authorized person asks for the patient's EKK and the receipt. Using EU and MEDULA the pharmacist searches the receipt record for the patient. When the record is found, UE sends a KDB request to KKEC through OYA. Prior to authentication, KDP is determined by KDPS. If KDPS is unreachable for some reason, the policy is set as the default policy by TOE in which authentication with the highest security level (fourth level) is performed. During authentication, symmetric and asymmetric card verifications are performed and certificates of EKKs are validated online using OCSP to make sure the EKKs are valid and they are not revocated. According to KDP, if necessary, biometric identifications of users are also carried out by TOE using fingerprint or fingervein biometrics of the patient and the pharmacist. After authentication, the KDBs, which are electronically signed by TOE, are sent to EU through OYA and then OYA forwards them to KDS where they are registered. If all the processes are completed successfully, the patient is given the medicines written in the prescription.

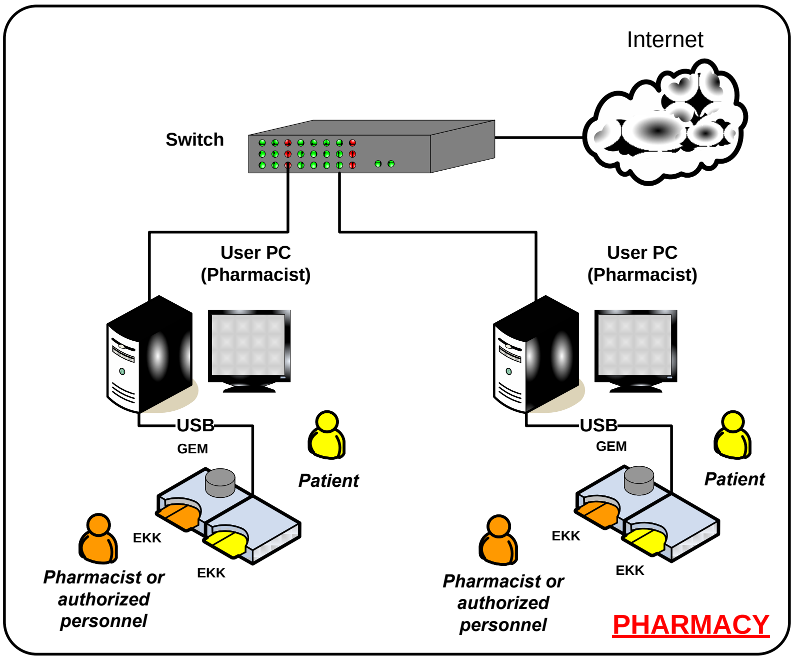

Figure 4. Pharmacy environment

1.4.1.3 Family Doctor Unit

As seen in figure 5 on the next page, the patient goes to a family doctor office/unit. First, the family doctor or his/her assistant asks for the patient's EKK. Using the web client on the family doctor's PC, the doctor initiates the operation. The application software requests KDBs from TOE through OYA. Prior to authentication, KDP is determined by KDPS. If KDPS is unreachable for some reason, the policy is set as the default policy by TOE in which authentication with the highest security level (fourth level) is performed. During authentication, symmetric and asymmetric card verifications are performed and certificates of EKKs are validated online using OCSP to make sure the EKKs are valid and they are not revocated. According to KDP, if necessary, biometric identifications of users are also carried out by TOE using fingerprint or fingervein biometrics of the patient and the doctor. After authentication, the KDBs, which are electronically signed by TOE, are sent to OYA. OYA forward them to KDS where they are checked and stored. If all the processes are completed successfully, an examination record is created.

Secondly, the doctor investigates the patient and decides whether a specialist doctor should see the patient. If it is not needed to send the patient to a specialist, the family doctor writes a prescription and sings it with his/her IK using KKEC. For this, TOE electronically signs the prescription by both the certificate in the doctor's IK and the certificate in GEM to guaranty non-repudiation of the operation.

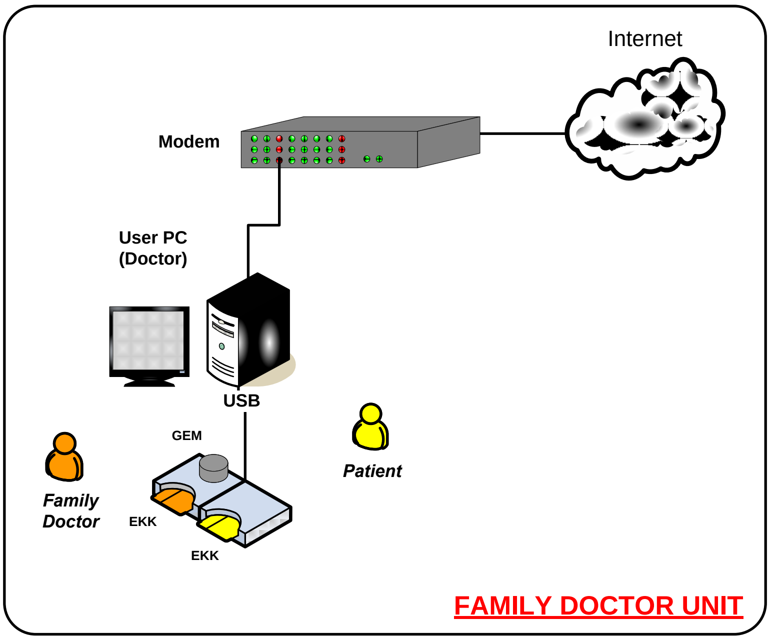

Figure 5. Family doctor environment

2. Conformance Claims

2.1 CC Conformance Claim

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 3.1, Revision 3, July 2009, conformant
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, Version 3.1, Revision 3, July 2009, conformant
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components,Version 3.1, Revision 3, July 2009, conformant

2.2 PP and Package Claim

2.2.1 Protection Profile (PP) Claim

No claim.

2.2.2 Package Claim

EAL 4 augmented (ALC_DVS.2).

This Security Target elaborated in conformance with “Common Criteria for Information Technology Security Evaluation, Version 3.1 rev 3” contains the IT security requirements of the TOE and specifies the functional and assurance security measures to meet the stated requirements.

2.3 Conformance Rationale

The assurance level of EAL 4+ (ALC_DVS.2) is sufficient for this type of TOE since it is intended to defend against attacks that can be made given the assumptions and the threats defined in chapter 3.

3. Security Problem Definition

This section includes the following:

- Secure usage assumptions; and
- Threats.

The information presented here provides the basis for the security objectives specified in chapter 4, the security functional requirements in chapter 6 and TOE summary specifications in chapter 7.

3.1 Assumptions

This section describes the assumptions that must be satisfied by the TOE environment.

3.1.1 Assumptions upon the development environment

A_DES.01 The designer issues and maintains a written procedure describing the security rules, and applies it in the development environment.

A_DES.02 The designer ensures protection of security relevant information involved in the design stage and during the software signature phase.

3.1.2 Assumptions upon the production environment

A_MAN.01 The manufacturer maintains a written procedure describing the security rules, and applies it in the production environment.

A_MAN.02 The manufacturer ensures protection of security relevant information involved in the manufacturing phase and the testing stage.

A_MAN.03 Security measures exist on the personal computer connected to KKEC to ensure protection of the PC from viruses and unwanted programs and secure transfer of the TOE relevant data over the internet.

3.1.3 Assumptions upon the initialization and maintenance environment

A_INIT.01 KECÖB maintains a written procedure describing the security rules, and applies it in pre-use and post-use environment.

A_INIT.02 KECÖB ensures protection of security relevant information involved in personalization, delivery, maintenance phase and end of life processes.

A_INIT.03 Security measures exist on the personal computer connected to KKEC to ensure protection of the PC from viruses and unwanted programs and secure transfer of the TOE relevant data over the internet.

A_INIT.04 EKKs are issued by an authorized association. This authorized association initializes each EKK such that symmetric and asymmetric keys are written in a securely created folder in the smartcard.

3.1.4 Assumptions upon the use environment

A_USE.01 Security measures exist on the personal computer connected to KKEC to ensure protection of the PC from viruses and unwanted programs.

A_USE.02 OYA, which the TOE communicates to, is always an authorized OYA.

3.2 Organisational Security Policies

No Organizational Security Policy.

3.3 Threats

The TOE as defined in Chapter 1 is required to counter the threats described hereafter; a threat agent wishes to abuse the assets either by functional attacks or by environmental manipulations, or by specific hardware manipulations, or by any other type of attacks.

3.3.1 Threat Agents

A threat agent to the TOE can be:

- **User:** A person who has received a KKEC in an authorized way and who wants to alter transaction data or:
    - To replace at least one of the internal TOE assets by fake ones.
    - To alter the TOE to use it in an unauthorized manner.
    - To tamper the TOE in order to obtain security relevant information.
- **Aggressor:** A person who has received a KKEC in an unauthorized way and wants to alter transaction data or:
    - To replace at least one of the internal TOE assets by fake ones.
    - To alter the TOE to use it in an unauthorized manner.
    - To tamper the TOE in order to obtain security relevant information.

3.3.2 Threats covered by the TOE

T_TECH Due to technical failure of some critical components, TOE security functions may not execute as expected as in its secure state. These failures may not make the TOE stop functioning completely but may affect its security.

T_PNTR By a threat agent, unauthorized opening of KKEC to obtain or modify security relevant data within TOE during the use stage.

3.3.3 Threats covered by the TOE and the environment

T_KKEC A threat agent may use a fake KKEC to obtain a TOE service in an unauthorized way.

T_SC A threat agent may use a fake EKK or IK to obtain a TOE service in an unauthorized way.

T_GSP A threat agent may imitate GSP to connect to TOE or modify the data between TOE and GSP in order to obtain a service in an unauthorized way.

T_HUBC A threat agent may use a fake HUBC to obtain PIN or biometric information of users or modify the data transferred by in order to obtain a service in an unauthorized way.

T_OCSPS A threat agent may imitate OCSPS or modify the data sent by OCSPS in order to obtain a service in an unauthorized way.

T_SPS A threat agent may imitate SPS or modify the software upgrade packets sent by SPS in order to modify TOE in an unauthorized way.

T_FRAUD A threat agent may use somebody else's valid EKK or IK to obtain a TOE service in an unauthorized way.

T_MNTR	By a threat agent, unauthorized local or remote monitoring of electromagnetic radiation emitted from KKEC or directly the data transferred between KKEC and other environmental components (GSP/OYA, HUBC and EKK/IK/GEM) to discover security relevant information during the use stage.
T_REPU	A user may repudiate an operation performed by the TOE with his/her approval.

4. Security Objectives

The security objectives of the TOE and its environment cover principally the following aspects:

- Integrity and confidentiality of assets,
- Protection of TOE and the associated documentation during development and production phases.

4.1 Security Objectives for the TOE

OT_INTEGRITY The TOE shall provide the means of detecting loss of integrity affecting security information stored or used by TOE.

OT_CONF The TOE shall ensure confidentiality of security relevant data during storage and use and also transferred between TOE and GSP/OYA/EKK/IK/HUBC.

OT_NON_REPU The TOE shall provide evidence for identity of origin, time of origin and location of origin for any document digitally signed by the TOE.

OT_USER_AUTH The TOE shall authenticate users at user authentication request coming from OYA/GSP.

OT_EKK_AUTH The TOE shall authenticate EKK/IK before during each user authentication.

OT_GEM_AUTH The TOE shall authenticate GEM during each user authentication.

OT_HUBC_AUTH The TOE shall authenticate HUBC during connection.

OT_GSP_AUTH The TOE shall authenticate GSP during connection.

OT_SOFT_AUTH The TOE shall authenticate software upgrade packets during remote upgrade.

OT_TEST The TOE shall be able to perform built-in and user initiated tests for security related functions to ensure correct operation of critical hardware components of KKEC and TOE logical functions.

OT_AUDIT The TOE shall create audit records for critical operations and detect security violations. TOE shall also protect audit records against possible data lost or data overflow.

4.2 Security Objectives for the Operational Environment

OE_USR_AWR Users shall be informed of their responsibility to protect their EKK/IK, and PIN and PUK information.

OE_SCARD Smartcards shall have security measures to ensure authentication of EKK/IK/GEM and cardholder by TOE; confidentiality of personal information and security relevant data. The cryptographic keys within smartcards shall be protected against unauthorized disclosure. The keys will be written in a secure folder.

OE_PROCED At each phase of its life cycle, the entity in charge of TOE shall issue and maintain a written procedure to apply them during this phase.

OE_PROTECT In development, production, initialization (pre-use) and maintenance (post use) phases, the entity in charge of TOE shall ensure protection of security relevant data.

OE_PC PC connected to KKEC shall be protected against viruses and other unwanted programs.

OE_HUBC HUBC shall have security measures to ensure authentication of HUBC by TOE and confidentiality of security relevant data transferred between TOE and HUBC. The symmetric key within HUBC shall be protected against unauthorized disclosure.

OE_GSP GSP shall have security measures to ensure authentication of GSP by TOE and confidentiality of security relevant data transferred between TOE and GSP. Also GSP shall protect the private key of its certificate against unauthorized disclosure.

OE_OYA OYA shall have security measures to ensure confidentiality of security relevant data transferred between TOE and OYA; and to protect itself from unauthorized modification.

OE_OCSPS Private key of OCSPS certificate shall be protected against unauthorized disclosure.

OE_SPS Private key of SPS certificate shall be protected against unauthorized disclosure.

OE_ACCESS Security relevant information shall be accessible only by authorized personnel.

OE_KKEC_RTC KKEC shall have a RTC (Real Time Clock) hardware module.

OE_KKEC_EMC KKEC shall be compatible to EN55022 standard class B type.

OE_GEM_PIN GEM PIN shall be written within flash memory of KKEC (in file system of TOE) in an encrypted form by KECÖB personnel during initialization.

4.3 Security Objectives Rationale

The Table-1 shows security objectives' relation to threats and assumptions. It demonstrates that at least one security objective is correlated to at least one threat or one assumption, and that each threat or each assumption is countered by at least one security objective.

	OT_INTEGRITY	OT_CONF	OT_NON_REPU	OT_USER_AUTH	OT_EKK_AUTH	OT_GEM_AUTH	OT_GSP_AUTH	OT_HUBC_AUTH	OT_SOFT_AUTH	OT_TEST	OT_AUDIT	OE_USR_AWR	OE_SCARD	OE_PROCED	OE_PROTECT	OE_PC	OE_HUBC	OE_GSP	OE_OYA	OE_OCSPS	OE_SPS	OE_ACCESS	OE_KKEC_RTC	OE_KKEC_EMC	OE_GEM_PIN
A_DES.01														✓								✓			
A_DES.02															✓							✓			
A_MAN.01														✓								✓			
A_MAN.02															✓							✓			
A_MAN.03																✓									
A_INIT.01														✓								✓			
A_INIT.02															✓							✓			
A_INIT.03																✓									
A_INIT.04													✓												
A_USE.01																✓									
A_USE.02																			✓						
T_TECH										✓	✓														
T_REPU			✓	✓		✓					✓												✓		
T_PNTR	✓	✓																							
T_MNTR		✓																						✓	
T_KKEC						✓					✓														✓
T_SC					✓						✓		✓												
T_GSP		✓					✓				✓							✓							
T_HUBC		✓						✓			✓						✓								
T_OCSPS	✓										✓									✓					
T_SPS	✓	✓							✓		✓										✓				
T_FRAUD				✓							✓	✓													

Table 1. Mapping of the security objectives to the assumptions and the threats

A_DES.01 This assumption is countered by OE_PROCED and OE_ACCESS environmental objectives. By OE_PROCED objective, in development stage the designer has to obey some security procedures that are maintained by the entity in charge of TOE development. OE_ACCESS objective ensures that security relevant information must be accessible only by authorized personnel.

A_DES.02 This assumption is countered by OE_PROTECT and OE_ACCESS environmental objectives. By OE_PROTECT objective, the designer has to protect the security relevant data during the development phase. OE_ACCESS objective ensures that security relevant information must be accessible only by authorized personnel.

A_MAN.01 This assumption is countered by OE_PROCED and OE_ACCESS environmental objectives. By OE_PROCED objective, in production stage the manufacturer has to obey some security procedures that are maintained by the entity in charge of TOE production. OE_ACCESS objective ensures that security relevant information must be accessible only by authorized personnel.

A_MAN.02 This assumption is countered by OE_PROTECT and OE_ACCESS environmental objectives. By OE_PROTECT objective, the manufacturer has to protect the security relevant data during the production phase. OE_ACCESS objective ensures that security relevant information must be accessible only by authorized personnel.

A_MAN.03 This assumption is countered by OE_PC objective.

A_INIT.01 This assumption is countered by OE_PROCED and OE_ACCESS environmental objectives. By OE_PROCED objective, in pre-use stage the KECÖB personnel have to obey some security procedures that are maintained by the entity in charge of TOE initialization and personalization. OE_ACCESS objective ensures that security relevant information must be accessible only by authorized personnel.

A_INIT.02 This assumption is countered by OE_PROTECT and OE_ACCESS environmental objectives. By OE_PROTECT objective, the KECÖB personnel have to protect the security relevant data during the pre-use phase. OE_ACCESS objective ensures that security relevant information must be accessible only by authorized personnel.

A_INIT.03 This assumption is countered by OE_PC objective.

A_INIT.04 This assumption is countered by OE_SCARD objective.

A_USE.01 This assumption is countered by OE_PC objective.

A_USE.02 This assumption is countered by OE_OYA objective.

T_TECH	This threat is especially applicable to “USE” phase of TOE life-cycle. The threat happens when some components of the hardware environment stop functioning as the time passes. This is dangerous because it might cause TOE go out of the secure state. This threat is countered by OT_TEST and OT_AUDIT objectives. OT_TEST objective is realized by providing start-up and user-invoked test functions, therefore it enables the TOE to detect a hardware failure. OT_AUDIT objective is for reporting a technical failure in a log file.
T_KKEC	This threat is especially applicable to “USE” phase of TOE life-cycle. The threat happens when a threat agent replaces the KKEC or modifies it so that it becomes counterfeit. This is dangerous because the TOE is embedded within KKEC. To do this the threat agent must have a valid GEM card and he/she must know the PIN of the GEM card. Since the GEM PIN is embedded within TOE (application software’s object code) the threat agent has to extract it from that object code. This threat is countered by OT_GEM_AUTH, OT_AUDIT and OE_GEM_PIN objectives. OT_GEM_AUTH objective makes unable to use the TOE without a valid GEM. Otherwise the fake KKEC will not be able to use GEM. Without a valid GEM the fake KKEC is useless. OT_AUDIT objective is for reporting an authentication failure regarding GEM. With the help of OE_GEM_PIN objective, the threat agent must either know PIN of a valid GEM or extract the PIN from the flash memory of KKEC.
T_SC	This threat is especially applicable to “USE” phase of TOE life-cycle. The threat happens when a threat agent inserts a fake EKK/IK to KKEC. This is dangerous because the TOE might be unable to use its security functions such as user authentication and the threat agent might get an unauthorized access. This threat is countered by OT_EKK_AUTH, OT_AUDIT and OE_SCARD objectives. OT_EKK_AUTH objective requires each EKK to be authenticated before any operation is performed. OT_AUDIT objective is for reporting smartcard authentication failures. Additionally, with OE_SCARD objective, EKK/IK/GEMs cannot be copied.
T_GSP	This threat is especially applicable to “USE” phase of TOE life-cycle. The threat happens when a threat agent tries to connect to TOE from a fake GSP. This threat is countered by OT_CONF, OT_GSP_AUTH, OT_AUDIT and OE_GSP objectives. OT_GSP_AUTH objective requires authentication of GSP during connections while OT_CONF objective necessitates a secure communication between TOE and GSP. OT_AUDIT objective is for reporting GSP connection and operation failures. Finally, with OE_GSP environment objective, it is assured that private key of GSP certificate is kept secret.
T_HUBC	This threat is especially applicable to “USE” phase of TOE life-cycle. The threat happens when a thread agent imitates HUBC or modifies the data sent by a real HUBC in order to obtain PIN or biometric information of users or obtain a TOE service in an unauthorized way. This threat is countered by OT_CONF, OT_HUBC_AUTH, OT_AUDIT and OE_HUBC objectives. OT_HUBC_AUTH objective requires authentication of HUBC during connections while OT_CONF objective necessitates a secure communication between TOE and HUBC.

OT_AUDIT objective is for reporting HUBC connection and operation failures. Finally, with OE_GSP environment objective, it is assured that the symmetric key of HUBC is kept secret.

T_OCSPS This threat is especially applicable to "USE" phase of TOE life-cycle. The threat happens when the TOE receives data packets singed by a fake OCSPS or modified by a thread agent. This threat is countered by OT_INTEGRITY, OT_AUDIT and OE_OCSPS objectives. OT_INTEGRITY objective requires means of detecting integrity of data sent by OCSP. OT_AUDIT objective is for reporting errors related to OCSP operations. Finally, by OE_OCSPS environment objective, it is assured that private key of OCSPS certificate is kept secret.

T_SPS This threat is especially applicable to "USE" phase of TOE life-cycle. The threat happens when the TOE receives software upgrade packets singed by a fake SPS or modified by a thread agent. This threat is countered by OT_INTEGRITY, OT_CONF, OT_SOFT_AUTH, OT_AUDIT and OE_SPS objectives. OT_INTEGRITY objective requires means of detecting integrity of software upgrade packets sent by SPS while OT_CONF objective necessitates confidentiality of software upgrade packets through secure communication between TOE and GSP/OYA. By OT_SOFT_AUTH objective, authorization of any software upgrade packet sent by SPS is obliged. OT_AUDIT objective is for reporting errors related to software upgrade operations. Finally, by OE_SPS environment objective, it is assured that private key of SPS certificate is kept secret.

T_FRAUD This threat is especially applicable to "USE" phase of TOE life-cycle. The threat happens when a threat agent steals someone's EKK/IK and uses it to claim a service in an unauthorized way. This threat is countered by OT_USER_AUTH, OT_AUDIT and OE_USR_AWR objectives. OT_USER_AUTH requires user authentication. OT_AUDIT objective is for reporting user authentication failures. Additionally, OE_USR_AWR objective requires that users are aware of the importance of confidentiality of their EKK/IK, PIN and PUK.

T_MNTR This threat is especially applicable to "PREUSE" and "USE" phases of TOE life-cycle. The threat is unauthorized monitoring of electromagnetic radiation or monitoring electrical cables of KKEC to discover security relevant data. This threat is countered by OT_CONF and OE_KKEC_EMC objectives. By OT_CONF objective encryption of the communication interfaces is satisfied. OE_KKEC_EMC objective requires EMC compatibility of the hardware environment according to EN55022 standard.

T_PNTR This threat is especially applicable to "PREUSE" and "USE" phases of TOE life-cycle. It is countered by OT_INTEGRITY and OT_CONF objectives. OT_INTEGRITY objective is to provide the means of detecting loss of integrity affecting security information stored within the device, OT_CONF is to ensure confidentiality of security relevant information that TOE manages during storage and use.

T_REPU This threat is especially applicable to "USE" phase of TOE life-cycle. The threat is repudiation of an operation performed by a user. This threat is countered by OT_NON_REPU, OT_USER_AUTH, OT_GEM_AUTH, OT_AUDIT and OE_KKEC_RTC objectives. OT_NON_REPU is to assure that TOE provides evidence of identity of origin, location of origin, and time of origin for each sign operation. To do this, OT_USER_AUTH objective helps to provide identity of origin, OT_GEM_AUTH objective helps to provide location of origin, OE_KKEC_RTC objective helps to provide time of origin. Additionally, OT_AUDIT objective is for logging each operation performed by TOE.

5. Extended Components Definition

No extended components (SFR or SAR).

6. Security Requirements

6.1 Security Functional Requirements

This chapter defines the security functional requirements for the TOE according to the functional requirements components drawn from the CC version 3.1 rev 3 Part 2.

6.1.1 Class FAU: Security Audit

6.1.1.1 Security Alarms (FAU_ARP.1)

FAU_ARP.1.1 The TSF shall take [assignment: *switching to OUT OF SERVICE mode*] upon detection of a security violation.

6.1.1.2 Audit Data Generation (FAU_GEN.1)

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

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

b) All auditable events for the [selection: not specified] level of audit; and

c) [assignment: *digital signature generation, digital signature verification failures, authentication of EKK/IK/GEM, EKK/IK/GEM authentication failures, authentication of user, user authentication failures, PIN verification failures, fingerprint verification failures, fingervein verification failures*]

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the ST, [assignment: *location of event*]

6.1.1.3 User Identity Association (FAU_GEN.2)

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

6.1.1.4 Security audit analysis (FAU_SAA.1)

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

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

a) Accumulation or combination of [assignment: *GEM authentication failure, PIN entry failure for GEM*] known to indicate a potential security violation;

b) [assignment: *no additional rules.]*

6.1.1.5 Audit Review (FAU_SAR.1)

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

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

6.1.1.6 Selectable Audit Review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to apply [assignment: *time order*] of audit data based on [assignment: *the record with the latest time of origin is displayed first*].

6.1.1.7 Guarantees of Audit Data Availability (FAU_STG.2)

FAU_STG.2.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.2.2 The TSF shall be able to [selection: *prevent*] unauthorized modifications to the stored audit records in the audit trail.

FAU_STG.2.3 The TSF shall ensure that [assignment: *keeping two copies in a non-volatile memory*] stored audit records will be maintained when the following conditions occur: [selection: *power down, failure*].

6.1.1.8 Prevention of Audit Data Loss (FAU_STG.4)

FAU_ STG.4.1 The TSF shall [selection: *overwrite the oldest stored audit records*] and [assignment: *no additional actions*] if the audit trail is full.

6.1.2 Class FCO: Communication

6.1.2.1 Enforced Proof of Origin (FCO_NRO.2)

FCO_NRO.2.1 The TSF shall enforce the generation of evidence of origin for transmitted [assignment: *KDB, KB, any XML document to be signed, upgrade software*] at all times.

FCO_NRO.2.2 The TSF shall be able to relate the [assignments: *originator identity, time of origin, location of origin*] of the originator of the information, and the [assignment: *GEM signature, card signature*] of the information to which the evidence applies.

FCO_NRO.2.3 The TSF shall provide capability to verify the evidence of origin of information to [selection: GSP, OYA, service participant, service requester] given [assignment: *no limitation*].

6.1.3 Class FCS: Cryptographic Support

6.1.3.1 Cryptographic Key Generation (FCS_CKM.1)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *Ephemeral DH key exchange algoritm for SSL/TLS communication, proprietary key exhange algorithm for OYA and HUBC communication defined in "Akıllı Kart Tabanlı Sosyal Güvenlik Sistemi Kripto Mimarisi" document revision 1.9, public key is extracted from certificate for RSA, proprietary key exchange algorithm for secure messaging during smartcard communication defined in "Akıllı Kart güvenli Mesajlaşma Protokolü Tanım Dökümanı" documet revision 1.0*] and specified cryptographic key sizes [assignment: *256-bit symmetric session key for AES, 2048-bit key for RSA, 128-bit session key for 3DES*] that meet the following: [assignment: *Standart specifications for Public Key Cryptography - IEEE P1363-2000: Annex-D, part D.5.1; Akıllı Kart Tabanlı Sosyal Güvenlik Sistemi Kripto Mimarisi revision 1.9: part 5.3.1, part 6.6.6 and part 6.6.7; Akıllı Kart güvenli Mesajlaşma Protokolü Tanım Dökümanı revision 1.0*].

6.1.3.2 Cryptographic Operation (FCS_COP.1)

FCS_COP.1.1 The TSF shall perform [assignment: *data encryption and decryption, secure hash (message digest) computation, Online X.509 certificate validation, cryptographic key agreement, communication using TLS (Transport Layer Security) version 1.0*] in accordance with a specified cryptographic algorithm [assignment: *3DES, AES, RSA, SHA, HMAC-SHA*] and cryptographic key sizes [assignment: *128 bits for 3DES, 256 bits for AES, 2048 bits for RSA, 256 bits for SHA, 256 bits for HMAC-SHA*] that meet the following: [assignment: *FIPS 46-3 (Data Encryption Standard); FIPS 197 (Advanced Ecryption Standard); NIST SP800-38A (Recommendation for Block Cipher Modes of Operation); NIST SP800-3C (Recommendation for Block Cipher Modes of Operation: The CCM mode for Authentication and confidentiality); PKCS #1 v2.1 (RSA Cryptography Standard); PKCS #15 v1.1 (Cryptographic Token Information Syntax Standard); FIPS 180-3 (Secure Hash Standard); FIPS 198-1 (The Keyed-Hash Message authentication Code); IEEE P1363-2000 (Standart specifications for Public Key Cryptography); ITU–T Recommendation Information Technology X.509 – Open systems Interconnection – The Directory: Public-Key and Attribute Certificate Frameworks; RFC 2560 (X.509 Internet Public Key Infrastructure Online Certificate Status Protocol-OCSP); RFC 5246 (The Transport Layer Security Protocol)*].

6.1.4 Class FDP: User Data Protection

6.1.4.1 Data Authentication with Identity of Guarantor (FDP_DAU.2)

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: *GEM PIN, KDB, prescription written by a doctor, any XML based document digitally signed by a user*].

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

6.1.5 Class FIA: Identification and Authentication

6.1.5.1 Authentication Failure Handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when [assignment: *3*] unsuccessful authentication attempts occur related to [assignment: *PIN verification, fingerprint verification, fingervein verification*].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [selection: *met*], the TSF shall [assignment: *cancel the current operation displaying an error message, create an audit log*].

6.1.5.2 Timing of Authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow [assignment: *accessing all GUI menus except device configuration setup menu and advanced fingerprint test menu*] 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.

6.1.5.3 Unforgeable Authentication (FIA_UAU.3)

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

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

6.1.5.4 Single Use Authentication Mechanism (FIA_UAU.4)

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to [assignment: *asymmetric authentication of EKK using 2048-bit RSA algorithm*].

6.1.5.5 Multiple Authentication Mechanism (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide [assignment: *asymmetric authentication of EKK, digital photo verification, PIN verification, fingerprint* *verification, fingervein verification*] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [assignment: *asymmetric authentication of EKK must be satisfied, symmetric authentication of GEM must be satisfied, digital photo verification must be satisfied, PIN verification must be satisfied, fingerprint verification (optionally) needs to be satisfied, fingervein verification may (optionally) need to be satisfied*].

6.1.5.6 Re-Authenticating (FIA_UAU.6)

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions [assignment: *at user identity verification request*].

6.1.5.7 Protected Authentication Feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide only [assignment: *displaying star "*" character for each PIN character entered; and not displaying fingerprint image or fingervein template on the screen during biometric entry*] while the authentication is in progress.

6.1.5.8 Timing of Identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow [assignment: *accessing all GUI menus except device configuration setup menu and advanced fingerprint test menu*] 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.

6.1.6 Class FPT: Protection of the TSF

6.1.6.1 Reliable Time Stamps (FPT_STM.1)

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

6.1.6.2 Inter-TSF Basic TSF Data Consistency (FPT_TDC.1)

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret [assignment: *TSF data types described in "GSP-KKEC Communication Protocol" document revision 1.0 for GSP/OYA communication, "AKIS-KEC Arayüz" document revision 3.0 for EKK/IK/GEM communication*] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use [assignment: *the interpretation rules to be applied by the TSF described in "GSP-KKEC Communication Protocol" document revision 1.0 for GSP/OYA communication , "AKIS-KEC Arayüz" document revision 3 for EKK/IK/GEM communication*] when interpreting the TSF data from another trusted IT product.

6.1.6.3 Testing of External Entities (FPT_TEE.1)

FPT_TEE.1.1 The TSF shall run a suite of tests [selection: *at the request of an authorised user, periodically during normal operation, at card insertions*] to check the fulfillment of [assignment: *fingerprint sensor, fingervein device*].

FPT_TEE.1.2 If the test fails, the TSF shall [assignment: *inform the user when any faulty condition occurs with fingerprint sensor and fingervein device; continue with fingerprint option if fingervein device fail to operate*].

6.1.7 Class FTP: Trusted Path/Channels

6.1.7.1 Inter-TSF Trusted Channel (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit [selection: *the TSF and another trusted IT product [refinement: GSP/OYA, HUBC]*] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [assignment: p*ersonal identity verification of user, digital sign and sign verification, remote software upgrade*].

6.2 Security Assurance Requirements

This chapter defines the assurance requirements. The security assurance requirement level for the TOE is EAL 4 augmented (ALC_DVS.2). According to CC version 3.1 revision 3 documentation part 3 “Security Assurance Components”, TOE security assurance requirements has the following assurance components listed in Table-2.

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 documents	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.2 Sufficiency of security measures
	ALC_LCD.1 Developer defined life-cycle model
	ALC_TAT.1 Well-defined development tools
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST Introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.1 Testing: Basic Design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
AVA: Vulnerability Assessment	AVA_VAN.3 Focused vulnerability analysis

Table 2 TOE Assurance Components

6.3 Security Requirements Rationale

6.3.1 Security Functional Requirements Rationale

In this section, TOE SFRs are mapped to TOE objectives. Table 3 shows TOE objectives countered by TOE SFRs. This section demonstrates that each SFR is correlated to at least one security objective, and that each security objective is countered by at least one SFR.

	OT_INTEGRITY	OT_CONF	OT_NON_REPU	OT_USER_AUTH	OT_EKK_AUTH	OT_GEM_AUTH	OT_GSP_AUTH	OT_HUBC_AUTH	OT_SOFT_AUTH	OT_TEST	OT_AUDIT
FAU_ARP.1						✓					✓
FAU_GEN.1				✓	✓	✓	✓	✓	✓	✓	✓
FAU_GEN.2			✓								✓
FAU_SAA.1						✓					✓
FAU_SAR.1											✓
FAU_SAR.3											✓
FAU_STG.2											✓
FAU_STG.4											✓
FCO_NRO.2			✓						✓		
FCS_CKM.1		✓									
FCS_COP.1	✓	✓	✓	✓	✓		✓	✓	✓		
FDP_DAU.2	✓										
FIA_AFL.1				✓							
FIA_UAU.1					✓	✓					
FIA_UAU.3					✓	✓					
FIA_UAU.4					✓	✓					
FIA_UAU.5				✓	✓	✓					
FIA_UAU.6					✓	✓					
FIA_UAU.7				✓							
FIA_UID.1				✓							
FPT_STM.1			✓								✓
FPT_TDC.1		✓									
FPT_TEE.1										✓	
FTP_ITC.1		✓									

Table 3 Mapping of TOE SFRs to TOE security objectives

OT_INTEGRITY This objective is satisfied by FCS_COP.1 and FDP_DAU.2 SFRs. FDP_DAU.2 requires the functions to generate and verify evidence that can be used as a guarantee of the validity of security relevant data stored within the TOE. FCS_COP.1 requires necessary cryptographic operations to support FDP_DAU.2.

OT_CONF This objective is satisfied by FCS_CKM.1, FCS_COP.1, FPT_TDC.1 and FTP_ITC.1 SFRs. FCS_CKM.1 is for how to generate session keys for secure communication with GSP, OYA, HUBC, EKK, IK and GEM. FCS_CKM.1 is for generating a session key. FCS_COP.1 requires necessary cryptographic operations to support encryption of the security relevant data that TOE manages during storage and use. FPT_TDC.1 is for providing data consistency between TOE and GSP/OYA/HUBC/EKK/IK/GEM. Finally, FTP_ITC.1 defines a trusted channel between them.

OT_NON_REPU This objective is satisfied by FAU_GEN.2, FCO_NRO.2, FCS_COP.1 and FPT_STM.1 SFRs. FCO_NRO.2 requires the functions to generate and verify evidence of origin for KDB, KB and any XML document signed by a user. FAU_GEN.2 is for generating an audit record for an action of an identified user. FCS_COP.1 requires necessary cryptographic operations to support FCO_NRO.2. FPT_STM.1 provides a reliable time stamp for each KDB, KB and any XML document signed by a user.

OT_USER_AUTH This objective is satisfied by FAU_GEN.1, FCS_COP.1, FIA_AFL.1, FIA_UAU.5, FIA_UAU.7 and FIA_UID.1 SFRs. FIA_UAU.5 and FIA_UAU.7 SFRs are the requirements for user authentication. FCS_COP.1 requires necessary cryptographic operations to support user authentication. FIA_AFL.1 SFRs are for describing user authentication failure situations and course of action that will be taken by TOE. FAU_GEN.1 is for generating an audit record in case of failure. Finally FIA_UID.1 is for the requirements describing the operations that can be performed by TOE before the user is identified.

OT_EKK_AUTH This objective is satisfied by FAU_GEN.1, FCS_COP.1, FIA_UAU.1, FIA_UAU.3, FIA_UAU.4, FIA_UAU.5 and FIA_UAU.6 SFRs. FIA_UAU.1, FIA_UAU.3, FIA_UAU.4, FIA_UAU.5 and FIA_UAU.6 SFRs are the requirements to provide EKK/IK authentication. FCS_COP.1 requires necessary cryptographic operations to support EKK/IK authentication. FAU_GEN.1 is for generating an audit record in case of failure.

OT_GEM_AUTH This objective is satisfied by FAU_ARP.1, FAU_GEN.1, FAU_SAA.1, FIA_UAU.1, FIA_UAU.3, FIA_UAU.4, FIA_UAU.5 and FIA_UAU.6 SFRs. FIA_UAU.1, FIA_UAU.3, FIA_UAU.4, FIA_UAU.5 and FIA_UAU.6 SFRs are the requirements to provide GEM authentication. By FAU_ARP.1 SFR, TOE get OUT OF SERVICE mode when GEM authentication fails. FAU_SAA.1 SFR is for monitoring GEM authentication failures.FAU_GEN.1 is for generating an audit record in case of failure.

OT_GSP_AUTH This objective is satisfied by FAU_GEN.1 and FCS_COP.1 SFRs. FCS_COP.1 SFR requires necessary cryptographic operations to support GEM authentication. FAU_GEN.1 is for generating an audit record in case of failure.

OT_HUBC_AUTH This objective is satisfied by FAU_GEN.1 and FCS_COP.1 SFRs. FCS_COP.1 SFR requires necessary cryptographic operations to support

HUBC authentication. FAU_GEN.1 is for generating an audit record in case of failure.

OT_SOFT_AUTH This objective is satisfied by FAU_GEN.1, FAU_NRO.2, and FCS_COP.1 SFRs. FCO_NRO.2 requires the functions to generate and verify evidence of origin for upgrade software signed by SPS. FCS_COP.1 requires necessary cryptographic operations to enable authentication of upgrade software. In case of failure TOE cancels the upgrade process and informs the user. FAU_GEN.1 is for generating an audit record in case of failure.

OT_TEST This objective is satisfied by FAU_GEN.1 and FPT_TEE.1 SFR. FPT_TEE.1 SFR is for testing of external entities (fingerprint sensor, fingervein device) of TOE. FAU_GEN.1 is for generating an audit record in case of failure.

OT_AUDIT This objective is satisfied by FAU_ARP.1, FAU_GEN.1, FAU_GEN.2, FAU_SAA.1, FAU_SAR.1, FAU_SAR.3, FAU_STG.2, FAU_STG.4 and FPT_STM.1 SFRs. FAU_ARP.1 defines the actions in case a potential security violation is detected. FAU_GEN.1 defines the level of auditable events, and specifies the list of data that shall be recorded in each record. FAU_GEN.2 defines how auditable events are associated to individual users. FAU_SAA.1 is for detecting security violations by monitoring the listed events in FAU_SAA.1 description. FAU_SAR.1 defines the capability to read audit information from the audit records. FAU_SAR.3 states that there will be methods of ordering based on time of origin for audit records. FAU_STG.2 defines the requirement for protecting audit records from unauthorized deletion and modification, FAU_STG.4 is the requirement for preventing data loss in case the audit trail is full. Finally, FPT_STM.1 SFR provides time of event for each audit record.

6.3.2 Security Functional Requirements Dependencies

The selected security requirements include related dependencies. Table 6 below provides a summary of the security functional requirements dependency analysis.

Component	Dependencies	Which is:
FAU_ARP.1	FAU_SAA.1	Included
FAU_GEN.1	FPT_STM.1	FPT_STM.1 included
FAU_GEN.2	FAU_GEN.1	Included
	FIA_UID.1	Included
FAU_SAA.1	FAU_GEN.1	Included
FAU_SAR.1	FAU_GEN.1	Included
FAU_SAR.3	FAU_SAR.1	Included
FAU_STG.2	FAU_GEN.1	Included
FAU_STG.4	FAU_STG.1	FAU_STG.2 Included FAU_STG.2 is hierarchical to FAU_STG.1
FCO_NRO.2	FIA_UID.1	Included
FCS_CKM.1	FCS_CKM.2 or FCS_COP.1	FCS_COP.1 Included
	FCS_CKM.4	Not Included
FCS_COP.1	FCS_CKM.4	Not Included
	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	FCS_CKM.1 included
FDP_DAU.2	FIA_UID.1	Included
FIA_AFL.1	FIA_UAU.1	Included
FIA_UAU.1	FIA_UID.1	Included
FIA_UAU.3	None	-
FIA_UAU.4	None	-
FIA_UAU.5	None	-
FIA_UAU.6	None	-
FIA_UAU.7	FIA_UAU.1	Included
FIA_UID.1	None	-
FPT_TDC.1	None	-
FPT_TEE.1	None	-
FTP_ITC.1	None	-

Table 6 Summary of TOE Security Functional Requirements Dependencies

As seen in table 6 above, FCS_CKM.4 security functional requirement is not included in this ST altough it is required by FCS_COP.1 and FCS_CKM.1 SFRs. This is because there is no need to erase keys since keys used for 3DES, AES, SHA and HMAC-SHA algorithms are always session keys and keys used for RSA are public key of certificate.Session keys are valid untill the end of session. After the session, they are not used anymore.

6.3.3 Security Assurance Requirements Rationale

The security assurance requirement level for the TOE is EAL 4 augmented (ALC_DVS.2). All SARs of the TOE, described in chapter 6.2, are defined by CC version 3.1 rev 3 documentation part 3 “Security Assurance Components”. Accordingly, there are six assurance classes. The assurance classes and the documents/evaluation evidence prepared for that assurance class are as follows:

Assurance Class	Document	
	Name	Description
Security Target Evaluation	KKEC_UY_ST.doc	Security Target
Development	KKEC_UY_FS.doc	Functional Specification
	KKEC_UY_ÜTT.doc	Design Description
	KKEC_UY_GMT.doc	Security Architecture Description
	Source Code	Implementation Representation
Tests	KKEC_UY_TKD.doc	Test Coverage and Depth
	KKEC_UY_TST.doc	Tests
Life-cycle Support	KKEC_UY_KY.doc	Configuration Management
	KKEC_UY_YD.doc	Life-cycle
	KKEC_UY_GA.doc	Development Tools
	KKEC_UY_GOG.doc	Development Environment Security
	KKEC_UY_TSL.doc	Delivery
Guidance Documents	KKEC_UY_KİK.doc	Installation and Operational Guidance
Vulnerability Assessment	-	This assurance class is under evaluator responsibility.

Table 7 Mapping of the documents to the security assurance classes

All SARs of the TOE, defined under these six assurance classes, are therefore satisfied by the documents (except Vulnerability Assessment) in table 7.

7. TOE Summary Specification

In this section, TOE security functions are to be explained; and related TOE SFR(s) met by the SF are to be listed.

Sign:

In "*Sign*" security function; hash value of a given data is calculated and the result is encrypted (signed) by a smartcard inserted to KKEC using private key of a user's certificate. SHA-256 algorithm is used for hash calculation and 2048-bit RSA algorithm is used for data encryption.

This function meets FCS_COP.1, FCO_NRO.2, FDP_DAU.2, FAU_GEN.2 and FPT_STM.1 SFRs.

Sign Verification:

In "*Sign Verification*" security function; after verification of sign certificate, signed data is decrypted by public key of this certificate and the result is compared with calculated hash value of given data. SHA-256 algorithm is used for hash calculation and 2048-bit RSA algorithm is used for decryption.

This function meets FCS_COP.1, FCO_NRO.2, FDP_DAU.2 and FAU_GEN.2 SFRs.

Data Encryption using AES Algorithm:

In "*Data Encryption using AES Algorithm*" security function; the data is encrypted using AES algorithm with a specified 256-bit key.

This function meets FCS_COP.1SFR.

Data Decryption using AES Algorithm:

In "*Data Decryption using AES Algorithm*" security function; the data is decrypted using AES algorithm with a specified 256-bit key.

This function meets FCS_COP.1SFR.

Secure GEM/EKK/IK Communication:

In "*Secure GEM/EKK/IK Communication*" security function; the smartcard inserted to the specified smartcard slot is set into secure operation mode. In case of failure an audit record is created. In secure communication, the data transferred between the TOE and any smartcard is in encrypted form. The encryption is done using 3DES algorithm with a specified 128-bit session key. A session key is generated by the TOE and smartcard before secure communication starts.

This function meets FCS_CKM.1, FCS_COP.1, FTP_ITC.1 and FAU_GEN.1 SFRs.

Remote Software Upgrade:

In "*Remote Software Upgrade*" security function; remote upgrade of TOE is done securely.

This function meets FCS_CKM.1, FCS_COP.1 and FAU_GEN.1 SFRs.

EKK Authentication using KSTB:

In "*EKK Authentication using KSTB*" security function; the TOE authenticates EKK using KSTB (Cardholder Unique Identifier).

This function meets FCS_COP.1, FIA_UAU.1, FIA_UAU.5, FIA_UAU.6 and FAU_GEN.1 SFRs.

EKK/IK Authentication using Asymmetric Method:

In "*EKK/IK Authentication using Asymmetric Method*" security function; the TOE authenticates EKK/IK.

This function meets FCS_COP.1, FIA_UAU.1, FIA_UAU.3, FIA_UAU.4, FIA_UAU.5, FIA_UAU.6 and FAU_GEN.1 SFRs.

GEM Authentication using Symmetric Method:

In "*GEM Authentication using Symmetric Method*" security function; the TOE authenticates GEM.

This function meets FIA_UAU.1, FIA_UAU.3, FIA_UAU.4, FIA_UAU.5, FIA_UAU.6, FAU_SAA.1, FAU_ARP.1 and FAU_GEN.1 SFRs.

Validation of GEM Sign Certificate:

In "*Validation of GEM Sign Certificate*" security function; the TOE validates GEM sign certificate using OCSP.

This function meets FCS_COP.1, FAU_SAA.1, FAU_ARP.1 and FAU_GEN.1 SFRs.

User Identification using PIN Verification Method:

In "*User Identification using PIN Verification Method*" security function; PIN entered by the user is compared with PIN available within the user's EKK. The comparison is done inside the smartcard and the result is returned to the TOE.

This function meets FIA_UAU.5, FIA_UAU.7, FIA_AFL.1, FIA_UID.1 and FAU_GEN.1 SFRs.

User Identification using Fingerprint Verification Method:

In "*User Identification using Fingerprint Verification Method*" security function; fingerprint data read from the fingerprint sensor is compared with the fingerprint data within the cardholder's EKK/IK. The comparison is done by TOE.

This function meets FIA_UAU.5, FIA_UAU.7, FIA_AFL.1, FIA_UID.1 and FAU_GEN.1 SFRs.

User Identification using Fingervein Verification Method:

In "*User Identification using Fingervein Verification Method*" security function; fingervein data read from the fingervein device is compared with the fingervein data within the cardholder's EKK/IK. The comparison is done by the fingervein device and a score is sent to TOE. TOE decides if they match.

This function meets FIA_UAU.5, FIA_UAU.7, FIA_AFL.1, FIA_UID.1 and FAU_GEN.1 SFRs.

User Identification using Digital Photo Inspection:

In "*User Identification using Digital Photo Inspection*" security function; the cardholder's digital image stored in his/her EKK/IK is displayed on the LCD screen of KKEC.

This function meets FIA_UAU.5 and FIA_UID.1 SFRs.

Secure GSP Communication:

In "*Secure GSP Communication*" security function; secure communication session is established between the TOE and GSP.

This function meets FCS_CKM.1, FCS_COP.1, FPT_TDC.1, FTP_ITC.1 and FAU_GEN.1 SFRs.

Secure OYA Communication:

In "*Secure OYA Communication*" security function; secure communication session is established between the TOE and OYA.

This function meets FCS_CKM.1, FCS_COP.1, FPT_TDC.1 and FTP_ITC.1 SFRs.

Secure HUBC Communication:

In "*Secure HUBC Communication*" security function; secure communication session is established between the TOE and HUBC.

This function meets FCS_CKM.1, FCS_COP.1, FPT_TDC.1 and FTP_ITC.1 SFRs.

Fingerprint Test:

In "*Fingerprint Test Method*" security function; TOE tests the operation of the fingerprint sensor and the software functions for fingerprint verification.

This function meets FPT_TEE.1 SFR.

Fingervein Test:

In "*Fingervein Test Method*" security function; TOE tests the operation of the fingervein device connected to KKEC externally and the software functions for fingervein verification.

This function meets FPT_TEE.1 SFR.

Review Audit Records:

In "*Review Audit Records*" security function; all audit records can be displayed by TOE on the device screen with "time of origin", "location of origin" and "identity of origin" attributes in time order.

This function meets FAU_SAR.1, FAU_SAR.3, FAU_STG.2, FAU_STG.4 and FPT_STM.1 SFRs.