TÜBİTAK BİLGEM

INFORMATICS AND INFORMATION SECURITY RESEARCH CENTER

YİTAL – SEMICONDUCTOR TECHNOLOGIES RESEARCH LABORATORY

NATIONAL SMARTCARD IC UKTÜM-H v7.0 WITH DES – 3DES v7.0, AES256 v7.0, RSA2048 v7.0 LIBRARIES AND WITH IC DEDICATED SOFTWARE

SECURITY TARGET

Revision No	06
Revision Date	08.07.2013
Document Code	UKTÜM-H_v7.0 ST
Developer	TUBITAK-BILGEM-UEKAE
Department	YITAL

Publication No	**Reason of Publication**	**Publication Date**
01	First Publication	19.11.2012
02	Modifications according to observation report 4	04.02.2013
03	Modifications according to observation report 7	19.02.2013
04	Modifications according to observation report 8.	07.06.2013
05	SEF modifications	20.06.2013
06	FDP_SDI modification	08.07.2013

CONTENTS:

LIST OF TABLES:

LIST OF FIGURES

Table 1. Abbreviations

AES	Advanced Encryption Standard
BİLGEM	Center of Research For Advanced Technologies of Informatics and Information Security
CC	Common Criteria
CMOS	Complementary Metal Oxide Semiconductor
CRC	Cyclic Redundancy Check
DES	Data Encryption Standard
DPA	Differential Power Analysis
EAL	Evaluation Assurance Level
HHNEC	Hua Hong NEC Company, Shanghai, China
IC	Integrated Circuit
ID	Identification
I/O	Input/Output
NVM	Non Volataile Memory
OS	Operating System
PP	Protection Profile
RAM	Random Access Memory
RNG	Random Number Generator
ROM	Read Only Memory
RSA	Rivest, Shanir, Adelmann public key encryption algorithm
SEF	Security Enforcing Function
SFR	Security Functional Requirement
SRAM	Static Random Access Memory
ST	Security Target
TSF	TOE Security Functions
TOE	Target of Evaluation
TÜBİTAK	Scientific and Technological Research Council of Turkey
UART	Universal Asynchronous Receiver Transmitter
UEKAE	National Research Institute of Electronics and Cryptology
YİTAL	Semiconductor Technology Research Laboratory

1. INTRODUCTION

1.1. Security Target Reference

The Security Target Reference is given in Table 2.

Table 2. The Security Target Reference

Name of Security Target Document	ST Version	ST Publication Date
Security Target Document of National Smartcard IC UKTÜM-H v7.0 with DES-3DES v7.0, AES256 v7.0, RSA2048 v7.0 libraries and with IC Dedicated Software	6	08.07.2013

This Security Target describes the TOE, intended IT environment, security objectives, security requirements (for the TOE), TOE security functions and all necessary rationale.

1.1.1.Operation Notation for Functional Requirements

There are four types of operations that can be applied on functional requirements. These are;

- **Assignment:** Regular letter in bracket
- **Selection:** Italic letter in bracket
- **Iteration:** Numbers in bracket
- **Refinement:** Regular letter with underline in bracket

1.2.TOE Reference

The TOE reference is given in Table 3

Table 3. TOE Reference

The TOE	TOE Version	TOE Publication Date
National Smartcard IC UKTÜM-H v7.0 with DES – 3DES v7.0, AES256 v7.0, RSA2048 v7.0 libraries and with IC Dedicated Software	7	19.11.2012

1.3.TOE Overview

TOE is a contact-based smartcard IC which is designed for security-based applications. TOE also includes IC Dedicated Software and DES-3DES v7.0, AES256 v7.0, RSA2048 v7.0 libraries. TOE is designed by Semiconductor Technology Research Laboratory (YİTAL) division under National Research Institute of Electronics and Cryptology (UEKAE) of TUBITAK-BILGEM and fabricated with HHNEC’s 0.25µm eFlash technology process. It is aimed that this smartcard IC is utilised as Turkish national ID Card and national Health Card where secrecy and security is an issue.

National Smartcard IC, UKTÜM-H v7.0 consists of an 8052-type microprocessor with a 256 Byte internal memory, a 10K Test ROM, three 64K Flash memory, an 8K Static RAM, and a True Random Number Generator. Furthermore, it is equipped with the hardware implementations of the RSA2048 , the DES-3DES and the AES ciphering algorithms.

UKTÜM-H v7.0 supports all requirements needed by smart card applications such as secure authentication, encryption/decryption, electronic signature, RSA2048 key generation. It is equipped with different security features including active shield protection, security sensors, distributed layout, encrypted data and code storage, power analysis and fault attack protection, random number generation.

Depending on the user requirements, an UKTÜM-H product can have three different flash memory usage configurations as follows: In the first configuration, user can load operating system into one 64K flash memory, and other two flash memories are used as user data area. In the second configuration, user can load operating system into two 64K flash memory, and he/she can use other one as user data area. In the third configuration, the user can use flash memories such as in the second configuration. The difference between second and third configuration is that the operating system loader software adds CheckSum values of operating system loaded to an 64K flash memory to the other 64K flash memory and vice versa. The entire configuration is done during the manufacturing and testing process of the TOE according to the choice of the user. All differences between the products of this TOE are realized by means of blocking without changing the hardware. Therefore, all products of this TOE are equal from hardware perspective. The flash memory usage configuration is done by setting the according value in the chip configuration page, which is not available to the user.

The Test ROM stores the IC Dedicated Software used to support testing of the TOE during production. The IC Dedicated Software also includes flash loader for downloading user software to NVM. The TOE includes hardware of UKTÜM-H v7.0 Smartcard IC, IC Dedicated Software, Flash memory access and user libraries of the DES-3DES, AES and RSA algorithms, and related documentation. The user or/and a subcontractor can download software to flash memory blocks.

The Smartcard Embedded Software, i.e. the operating system and applications are not part of the TOE.

UKTÜM-H v7.0 communicates with the outer environment through a smartcard reader in accordance with ISO/IEC 7816-3 protocol. Smartcard IC is designed to be resistant against power and fault attacks. In addition, it is equipped with security sensors which sense physical attacks and environmental operating conditions.

The smartcard IC UKTÜM-H v7.0 is developed in order to be used as national ID card. It aims to ensure EAL 5+ assurance level of CC and to be a national choice for smart card ICs on the market in terms of functinality, performance and security measures.

1.4.TOE Description

1.4.1.Scope of the TOE

The TOE includes

- Smartcard IC UKTÜM-H v7.0 Hardware
- DES-3DES v7.0, AES256 v7.0, RSA2048 v7.0 Crypto Access Libraries
- IC Dedicated Software,
- Flash Memory Access Library,
- User Guidance Document : Security Requirements for Operating System

The operating system is not a part of the TOE.

1.4.2.Physical Scope of the TOE

UKTÜM-H v7.0 IC is a contact-based smartcard IC which is designed and developed for security-based applications. It is designed by ASIC design team of YİTAL using EF250 0.25µm e-Flash CMOS process technology and design library of HHNEC. The smartcard IC is fabricated in the fab of HHNEC.

The block diagram of UKTÜM-H v7.0 smartcard IC is shown in Figure 1. The hardware of the security IC consists of

- 8052-type microprocessor with 256B internal RAM,

- 10K Test ROM storing IC Dedicated Software,

- 8K SRAM for volatile data storage,

- 3 x 64K Flash memory for non-volatile storage,

- RSA2048 crypto algorithm block,

- DES-3DES crypto algorithm block,

- AES crypto algorithm block,

- SHA-256 coprocessor block,

- UART block ensuring the communication between IC and card reader according to ISO/IEC 7816-3 protocol,

- Cyclic Redundancy Check module giving the opportunity to calculate 16 bit check sum according to ISO 3309 standard.

- Random Number Generator block producing true random numbers,

- Regulator converting external power supply of 5V to an internal supply of 2.5V,

- On Chip Oscillator which produces internal clock signal,

- Security sensors for sensing/preventing physical attacks,

- Reset Circuitry controlling the internal reset signal production according to RESET input and security sensor outputs.

Security Sensors subsystem includes the clock frequency sensor, the internal and external supply voltage sensors and the temperature sensor which sense the operating environment. These sensors cause the smartcard IC to enter to reset state when detected environmental conditions are out of specified ranges.

The active shield and the countermeasures against the fault attacks are also parts of security circuits:

The active shield, which consists of metal lines, covers the surface of the IC and prevents the attacker from probing and acquiring any useful data. In case of sensing a short-circuit or an open-circuit on the active shield the smartcard IC enters to reset state.

The TOE enters to reset state when it detects that the contents of the critical registers ensuring the proper operation of TOE are corrupted due to fault attacks.

The crypto modules of the TOE has been designed to be resistant against SPA and DPA attacks. The microprocessor of the TOE is equipped with additional countermeasures against power analysis attacks.

SHA-256 coprocessor block is not available to the user and it is out of scope of the TOE.

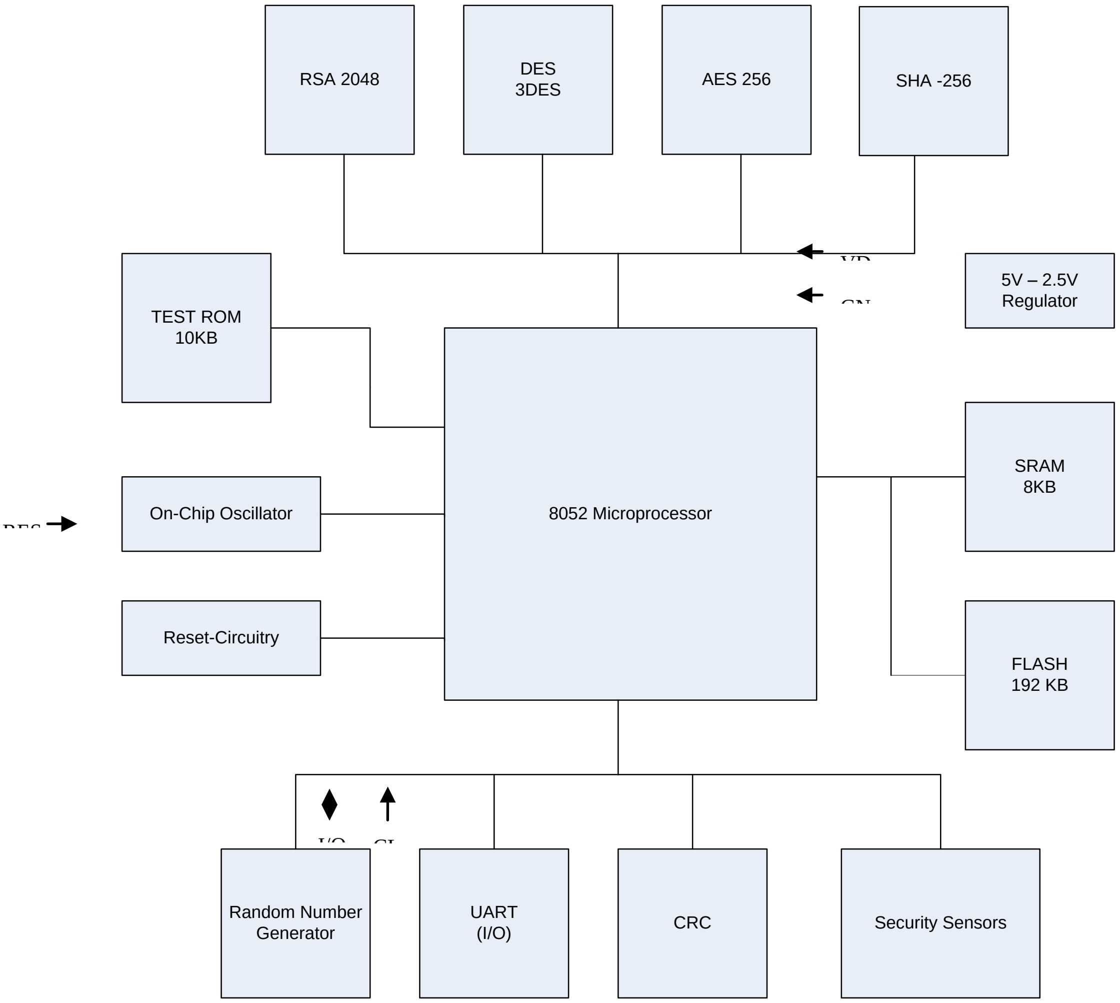

Figure 1. Smartcard IC Block Diagram

1.4.3. Interfaces of the TOE

· The entire surface of the IC constitutes the physical interface of the TOE to the external environment.

· CLK, RESET, I/O, VDD and GND pads of the IC constitute the electrical interface of the TOE to the external environment.

· The I/O pad of the IC constitutes the data input/output interface of the TOE to the external environment.

· The instruction set of the TOE and Special Function Registers controlling the hardware of the TOE constitute TOE's interface to the software environment.

· The flash memory access library constitutes the interface of the TOE to flash memory access operations.

· The RSA2048 library constitutes the interface of the TOE to the RSA calculations.

· The DES-3DES library constitutes the interface of the TOE to the DES-3DES calculations.

· The AES library constitutes the interface of the TOE to the AES calculations.

1.4.4.Logical Scope of the TOE

The operating system software which is loaded to the 64K/128K Flash block of the TOE uses 8051 instruction set to operate the smartcard IC hardware. During the TOE development, driver softwares for accessing to special modules such as Flash memory and crypto blocks have also been developed. During the development of the operating system, Flash memory access library and, to utilize the cryptographic operations, the RSA library, the AES library, the DES and 3DES library are given to the operating system developer as source codes. The subroutines codes needed to be called as indicated in the user guidance document 'Security Requirements for Operating System' are also provided to the OS developer.

The Test ROM includes IC dedicated self test and initialisation routines and also the flash loader. The self test software performs the operations such as the initial test after the manufacturing of the IC. Since the self test software is deactivated after manufacturing, it is not possible to access it by the operating system. The flash loader is used by the user or/and by a subcontractor for downloading user software to Flash Memory. For these cases and whenever the user has finalized his SW-download, the user is obligated to lock the Flash loader. The final locking of the FL results in a permanent deactivation of the Flash Loader. This means that once being in the locked status, the Flash Loader cannot be reactivated anymore.

1.4.5.Life Cycle

The design and manufacturing life cycle of UKTÜM-H v7.0 Smartcard IC is given in Figure 2.

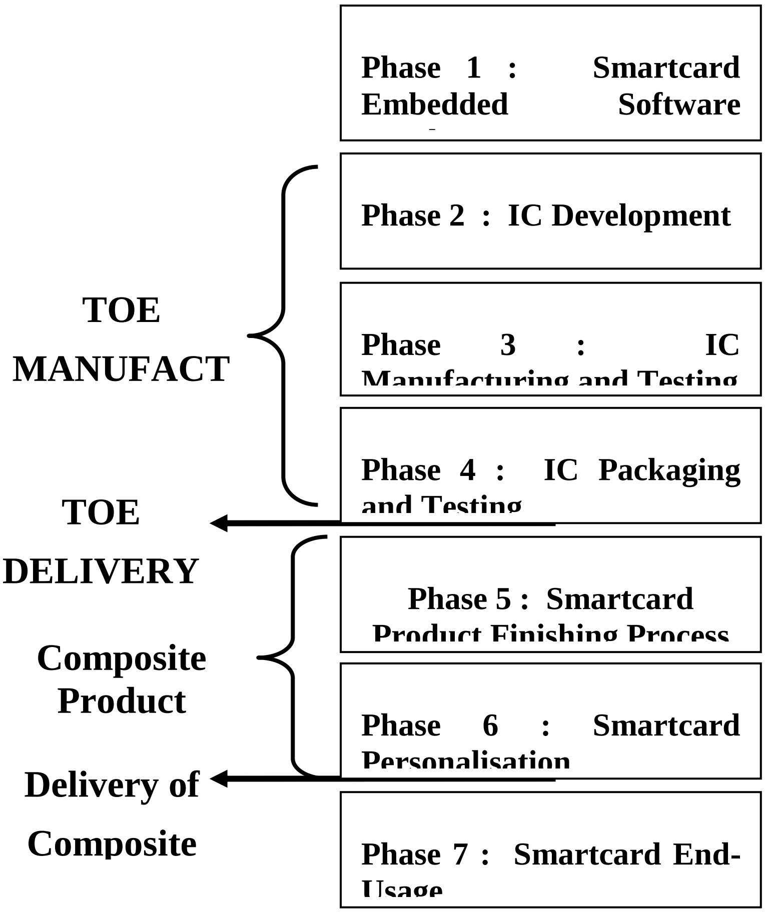
Phase 1 : Smartcard Embedded Software
Phase 2 : IC Development
Phase 3 : IC Manufacturing and Testing
Phase 4 : IC Packaging and Testing
Phase 5 : Smartcard Product Finishing Process
Phase 6 : Smartcard Personalisation
Phase 7 : Smartcard End-Usage
TOE MANUFACT
TOE DELIVERY
Composite Product
Delivery of Composite

Figure 2. **Life Cycle of the Composite Product**

Phase 1: Smartcard Embedded Software Development

In this phase the **Smartcard Embedded Software Developer** is in charge of

· the smartcard embedded software development and
· the specification of IC pre-personalisation requirements.

Since the operating system of the smartcard IC is developed in this phase, this phase will be out of the scope of the ST.

Phase 2: IC Development

In this phase, the **IC Designer**
· designs the IC,
· develops IC Dedicated Software,
· provides information, software or tools to the Smartcard Embedded Software Developer.
The information, software and tools given to the Smartcard Embedded Software Developer are flash memory access driver, crypto hardware driver, RNG test software and related documents about the UKTÜM-H v7.0
From the IC design and IC Dedicated Software , **the IC Designer** constructs the Security IC database, necessary for the IC photomask fabrication.

Phase 3: IC Manufacturing and Testing

In this phase, the **IC Manufacturer** is responsible for
· producing the IC through three main steps: IC manufacturing, IC testing, and IC pre-personalisation.
The **IC Mask Manufacturer**
· generates the masks for the IC manufacturing based upon an output from the smartcard IC database.
The security IC is manufactured with HHNEC's 0.25µm e-Flash process technology. When the manufacturing process is completed, wafer level tests are performed and the serial number which is specific for each individual chip is written on to the Flash memory of the chips passing the tests. This operation is performed through the IC Dedicated Software residing in the Test ROM. Next, after completing the operations regarding the phase 3, the circuit goes from self test-inializing mode to operating system load mode.

Phase 4: IC Packaging and Testing

In this phase, the **IC Packaging Manufacturer** is responsible for the IC packaging and testing.

At the end of the manufacturing stage, IC Manufacturer sends wafers to **IC Packaging Manufacturer** for packaging. There, wafers are diced and separated into individual chips. These individual chips are placed into smartcard modules and wire bonding operation is performed. **TOE is delivered in form of smartcard modules at the end of Phase 4 .**

Phase 5: Smartcard Product Finishing Process

In this phase, the operating system is loaded to TOE, and the Smartcard Product Developer do the smartcard product finishing process and testing.

Phase 6 : Smartcard Personalisation

In this phase, **the Personaliser** is responsible for the smartcard personalisation and final tests.

Phase 7 : Smartcard End-usage

In this phase, **the Smartcard Issuer** is responsible for the smartcard product delivery to the smartcard end-user, and the end of life process.

The Security IC Embedded Software is developed outside the TOE development in Phase 1. The TOE is developed in Phase 2 and produced in Phase 3. Then the TOE is packaged in Phase 4 and delivered in form of packaged products.

1.4.6.Life-Cycle versus Scope and Organisation of this ST

In this ST, the term "TOE Delivery" is uniquely used to indicate after Phase 4 (or before Phase 5) since the TOE is delivered in form of packaged products.

This ST uniquely uses the term "TOE Manufacturer" which includes the following roles:

- the IC Developer (Phase 2),
- the IC Manufacturer (Phase 3) and
- the IC Packaging Manufacturer (Phase 4)

Hence the "TOE Manufacturer" comprises all roles beginning with Phase 2 and before "TOE delivery". Starting with "TOE Delivery" another party takes over the control of the TOE. **This ST defines assurance requirements for the TOE's development and production environment up to "TOE Delivery". (Phase 2-4)**

The ST uniquely uses the term "Composite Product Manufacturer" which includes all roles (outside TOE development and manufacturing) except the End-consumer as user of the Composite Product which are the following:

- Security IC Embedded Software development (Phase 1)
- the Composite Product Manufacturer (Phase 5) and
- the Personaliser (Phase 6).

The TOE can be delivered at the end of phase 4 in any form of complete module, package or in an IC case or in bare dies. Since the pre-personalization steps are completed at the end of phase 3, the TOE is finished and the extended test features are removed.

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.

- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 3.1, Revision 3, July 2009,extended.

- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1, Revision 3, July 2009,conformant.

2.2.PP Conformance Claim

In this ST, TOE does not claim any conformance to a protection profile. But it uses the following Protection Profile (PP) as a guidance:

Security IC Platform Protection Profile, Version 1.0, 15.06.2007 Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0035.

2.3.Package Conformance Claim

Security Level: EAL 5+ (AVA_VAN.5).

2.4.Conformance Rationale

CC Conformance Claim Rationale: The latest version of the CC is used in the development and the evaluation of the TOE.

Package Conformance Claim: An assurance requirement of EAL5 is required for this type of TOE since

i. Semiformal design assurance is needed in the application context of the TOE such as the use of the TOE as a national ID or health-card.

ii. In order to provide a meaningful level of assurance that the TOE provides an adequate level of defense against sophisticated attacks, the evaluators should have access the hardware and the software source codes.

The augmented component AVA_VAN.5 is the highest level for AVA_VAN. EAL5 package is augmented with AVA_VAN.5 since the TOE is intended to defend against sophisticated attacks.

3. SECURITY PROBLEM DEFINITION

This section includes the following:

- Threats.
- Secure usage assumptions; and
- Organizational security policies;

This information provides the basis for the Security Objectives specified in Section 4, the Security Functional Requirements for the TOE specified in Sections 6.1 and the TOE Security Assurance Requirements specified in Section 6.2.

3.1.Threats

The following explanations help to understand the focus of the threats and objectives defined below. For example, certain attacks are only one step towards a disclosure of assets, others may directly lead to a compromise of the application security.

- Manipulation of data (which may comprise any data, including code, stored in or processed by the Security IC) means that an attacker is able to alter a meaningful block of data. This should be considered for the threats T.Malfunction, T.Phys-Manipulation and T.Abuse-Func.

- Manipulation of the TOE means that an attacker is able to deliberately deactivate or otherwise change the behaviour of a specific function in a manner which enables exploitation. This should be considered for the threat T.Malfunction, T.Phys-Manipulation and T.Abuse-Func.

- Disclosure of data (which may comprise any data, including code, stored in or processed by the Security IC) means that an attacker is realistically able to determine a meaningful block of data. This should be considered for the threats T.Leak-Inherent, T.Phys-Probing, T.Leak-Forced and T.Abuse-Func.

The cloning of the functional behaviour of the Security IC on its physical and command interface is the highest level security concern in the application context. The cloning of that functional behaviour requires to

(i) develop a functional equivalent of the Security IC Embedded Software,

(ii) disclose, interpret and employ the secret User Data stored in the TOE, and

(iii) develop and build a functional equivalent of the Security IC using the input from the previous steps.

The Security IC is a platform for the Security IC Embedded Software which ensures that especially the critical User Data are stored and processed in a secure way. The Security IC Embedded Software must also ensure that critical User Data are treated as required in the application context.

As a result the threat “cloning of the functional behaviour of the Security IC on its physical and command interface” is averted by the combination of mechanisms which split into those being evaluated according to this ST (Security IC) and those being subject to the evaluation of the Security IC Embedded Software or Security IC and the corresponding personalisation process. Therefore, functional cloning is indirectly covered by the security concerns and threats described below.

The Security IC Embedded Software may be required to contribute to averting the threats. At least it must not undermine the security provided by the TOE. Therefore the Security IC Embedded Software must ensure security against high attack potential which means an assurance component of AVA_VAN.5

The above security concerns are derived from considering the operational usage by the end-consumer (Phase 7) since

- Phase 1 and the Phases from TOE Delivery (Phase 5) up to the end of Phase 6 are covered by assumptions and

- the development and production environment starting with Phase 2 up to TOE Delivery are covered by an organisational security policy.

The TOE's countermeasures are designed to avert the threats described below.

3.1.1.Threats defined in the TOE

In Table 4, standard threats defined in the TOE are given and they are explained below

Table 4. Threats defined in the TOE

	THREAT NAME	BRIEF DESCRIPTION
1.	T. Leak-Inherent	Inherent Information Leakage
2.	T. Phys-Probing	Physical Probing
3.	T. Phys-Manipulation	Physical Manipulation
4.	T. Malfunction	Malfunction due to Environmental Stress
5.	T. Leak_Forced	Forced Information Leakage
6.	T. Abuse-Func	Abuse of Functionality
7.	T. RND	Deficiency of Random Numbers

T. Leak-Inherent : Inherent Information Leakage

An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential User Data as part of the assets. No direct contact with the Security IC internals is required here. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements. One example is Differential Power Analysis (DPA). This leakage may be interpreted as a covert channel transmission but is more closely related to measurement of operating parameters, which may be derived either from direct (contact) measurements or measurement of emanations and can then be related to the specific operation being performed.

T. Phys-Probing: Physical Probing

An attacker may perform physical probing of the TOE in order to

- disclose User Data,
- disclose/reconstruct the Security IC Embedded Software or
- disclose other critical information about the operation of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software.

Physical probing requires direct interaction with the Security IC internals. Techniques commonly employed in IC failure analysis and IC reverse engineering efforts may be used. Before that hardware security mechanisms and layout characteristics need to be identified. Determination of software design including treatment of User Data may also be a prerequisite. This pertains to “measurements” using galvanic contacts or any type of charge interaction whereas manipulations are considered under the threat “Physical Manipulation (T.Phys-Manipulation)”. The threats “Inherent Information Leakage (T.Leak-Inherent)” and “Forced Information Leakage (T.Leak-Forced)“ may use physical probing but require complex signal processing as well.

T. Phys-Manipulation: Physical Manipulation

An attacker may physically modify the Security IC in order to

- modify User Data,
- modify the Security IC Embedded Software,
- modify or deactivate security services of the TOE, or
- modify security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software.

The modification may be achieved through techniques commonly employed in IC failure analysis and IC reverse engineering efforts. The modification may result in the deactivation of a security feature. Before that hardware security mechanisms and layout characteristics need to be identified. Determination of software design including treatment of User Data may also be a pre-requisite. Changes of circuitry or data can be permanent or temporary. In contrast to malfunctions (refer to T.Malfunction) the attacker requires to gather significant knowledge about the TOE's internal construction here.

T. Malfunction: Malfunction due to Environmental Stress

An attacker may cause a malfunction of TOE Security Functions (TSF) or of the Security IC Embedded Software by applying environmental stress in order to

- modify security services of the TOE or
- modify functions of the Security IC Embedded Software
- deactivate or affect security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software.

This may be achieved by operating the Security IC outside the normal operating conditions. The modification of security services of the TOE may e.g. affect the quality of random numbers provided by the random number generator up to undetected deactivation when the random number generator does not produce random numbers and the Security IC Embedded Software gets constant values. In another case, errors are introduced in executing the Security IC Embedded Software. To exploit this, an attacker needs information about the functional operation, e.g., to introduce a temporary failure within a register used by the Security IC Embedded Software with light or a power glitch.

T. Leak_Forced: *Forced Information Leakage*

An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential User Data as part of the assets even if the information leakage is not inherent but caused by the attacker.

This threat pertains to attacks where methods described in “Malfunction due to Environmental Stress” (refer to T.Malfunction) and/or “Physical Manipulation” (refer to T.Phys-Manipulation) are used to cause leakage from signals which normally do not contain significant information about secrets.

T. Abuse-Func: Abuse of Functionality

An attacker may use functions of the TOE which may not be used after TOE Delivery in order to

- disclose or manipulate User Data,
- manipulate (explore, bypass, deactivate or change) security services of the TOE or
- manipulate (explore, bypass, deactivate or change) functions of the Security IC Embedded Software or
- enable an attack disclosing or manipulating the User Data or the Security IC Embedded Software.

T. RND: Deficiency of Random Numbers

An attacker may predict or obtain information about random numbers generated by the TOE security service for instance because of a lack of entropy of the random numbers provided.

An attacker may gather information about the random numbers produced by the TOE security service. Because unpredictability is the main property of random numbers this may be a problem in case they are used to generate cryptographic keys. Here the attacker is expected to take advantage of statistical properties of the random numbers generated by the

TOE. Malfunctions or premature ageing are also considered which may assist in getting information about random numbers.

3.2.Assumptions

The assumptions applied in this ST is given in Table 5 and explained below.

Table 5. Assumptions applied in this ST

	ASSUMPTION	BRIEF DESCRIPTION
1.	A.Process-Sec-IC	Protection during Packaging, Finishing and Personalisation (Phases 5-6)
2.	A.Plat-Appl	Usage of Hardware Platform
3.	A.Resp-Appl	Treatment of User Data
4.	A.Key-Function	Usage of key dependent Functions

A.Process-Sec-IC : *Protection during Finishing and Personalisation (Phases 5 – 6)*

Security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use). This means that the Phases after TOE Delivery are assumed to be protected appropriately.

A.Plat-Appl : *Usage of Hardware Platform*

The Security IC Embedded Software is designed so that the requirements from the following documents are met:

- UKTÜM-H v7.0 Security Requirements for Operating System
- Findings of the TOE evaluation reports relevant for the Security IC Embedded Software as referenced in the certification report.

Since particular requirements for the Security IC Embedded Software are not clear before considering a specific attack scenario during vulnerability analysis of the Security IC (AVA_VAN), a summary of such results is provided in the document "ETR for composite evaluation" (ETR-COMP). This document can be provided for the evaluation of the composite product. The ETR-COMP may also include guidance for additional tests being required for the combination of hardware and software. The TOE evaluation must be completed before evaluation of the Security IC Embedded Software can be completed. The TOE evaluation can be conducted before and independent from the evaluation of the Security IC Embedded Software.

A.Resp-Appl : *Treatment of User Data*

All User Data are owned by Security IC Embedded Software. Therefore, it must be assumed that security relevant User Data (especially cryptographic keys) are treated by the Security IC Embedded Software as defined for its specific application context.

The application context specifies how the User Data shall be handled and protected. The Security IC can not prevent any compromise or modification of User Data by malicious Security IC Embedded Software. The assumption A.Resp-Appl ensures that the Security IC Embedded Software follows the security rules of the application context. When defining the Protection Profile or Security Target for the evaluation of the Security IC Embedded Software appropriate threats must be defined which depend on the application context. These security needs are condensed in this assumption (A.Resp-Appl) which is very general since the application context is not known and the evaluation of the Security IC Embedded Software is not covered by this Security Target.

A.Key-Function: Usage of key dependent Functions

Key-dependent functions (if any) shall be implemented in the Smartcard Embedded Software in a way that they are not susceptible to leakage attacks (as described under T.Leak-Inherent and T.Leak-Forced).

Note that here the routines which may compromise keys when being executed are part of the Smartcard Embedded Software. In contrast to this the threats T.Leak-Inherent and T.Leak-Forced address

- The cryptographic routines which are part of the TOE and
- The processing of using data including cryptographic keys.

3.3. Organisational Security Policies

The Organisational Security Policies applied in this ST is given in Table 6 and explained below.

Table 6. Policies applied in this ST

	POLICY	BRIEF DESCRIPTION
1.	*P. Process-TOE*	*Protection during TOE Development and Production*
2.	*P. Add-Functions*	*Additional Specific Security Functionality*

P. Process-TOE: Protection during TOE Development and Production

The TOE Manufacturer must ensure that the development and production of the Smartcard Integrated Circuit (Phase 2 - 4) is secure so that no information is unintentionally made available for the operational phase of the TOE. For example, the confidentiality and integrity of design information and test data shall be guaranteed; access to samples, development tools and other material shall be restricted to authorised persons only; scrap will be destroyed etc. This not only pertains to the TOE but also to all information and material exchanged with the developer of the Smartcard Embedded Software and therefore especially to the Smartcard Embedded Software itself. This includes the delivery (exchange) procedures for Phase 1 and the Phases after TOE Delivery as far as they can be controlled by the TOE Manufacturer.

An accurate identification must be established for the TOE. This requires that each instantiation of the TOE carries this unique identification. The accurate identification is introduced at the end of the production test in phase 3. Therefore the production environment must support this unique identification.

P. Add-Functions: Additional Specific Security Functionality

The TOE shall provide the following specific security functionality to the Smartcard Embedded Software:

- Data Encryption Standard (DES)
- Triple Data Encryption Standard (3DES)
- Advanced Encryption Standard (AES256)
- Rivest-Shamir-Adleman (RSA1024, RSA2048)
- Cryptographic Key Generation or RSA2048 algorithm

4. SECURITY OBJECTIVES

4.1.Security Objectives for the TOE

The user have the following standard high-level security goals related to the assets:

SG1: Maintain the integrity of User Data and of the Security IC Embedded Software (when being executed/processed and when being stored in the TOE's memories) as well as,

SG2: Maintain the confidentiality of User Data and of the Security IC Embedded Software (when being processed and when being stored in the TOE's memories),

SG3: Maintain the correct operation of the security services provided by the TOE for the Security IC Embedded Software.

SG4: Provide true random number generator.

The additional high-level security considerations are refined below by defining security objectives as given above. They are listed in Table 7 and explained below:

Table 7. Objectives for the TOE

	OBJECTIVE	BRIEF DESCRIPTION
1.	O. Leak-Inherent	Protection against Inherent Information Leakage
2.	O. Phys-Probing	Protection against Physical Probing
3.	O. Phys-Manipulation	Protection against Physical Manipulation
4.	O. Malfunction	Protection against Malfunction
5.	O. Leak_Forced	Protection against Forced Information Leakage
6.	O. Abuse-Func	Protection against Abuse of Functionality
7.	O. Identification	TOE Identification
8.	O. RND	Protection against Deficiency of Random Numbers

9.	O.Add_Functions	Additional Specific Security Functions

O.Leak-Inherent: Protection against Inherent Information Leakage

The TOE must provide protection against disclosure of confidential data stored and/or processed in the Security IC

- by measurement and analysis of the shape and amplitude of signals (for example on the power, clock, or I/O lines) and
- by measurement and analysis of the time between events found by measuring signals (for instance on the power, clock, or I/O lines).

This objective pertains to measurements with subsequent complex signal processing whereas O.Phys-Probing is about direct measurements on elements on the chip surface.

O.Physical-Probing: *Protection against Physical Probing*

The TOE must provide protection against disclosure of User Data, against the disclosure/reconstruction of the Security IC Embedded Software or against the disclosure of other critical information about the operation of the TOE. This includes protection against

- measuring through galvanic contacts which is direct physical probing on the chips surface except on pads being bonded (using standard tools for measuring voltage and current) or
- measuring not using galvanic contacts but other types of physical interaction between charges (using tools used in solid-state physics research and IC failure analysis)

with a prior reverse-engineering to understand the design and its properties and functions.

The TOE must be designed and fabricated so that it requires a high combination of complex equipment, knowledge, skill, and time to be able to derive detailed design information or other information which could be used to compromise security through such a physical attack.

O. Malfunction: *Protection against Malfunctions*

The TOE must ensure its correct operation.

The TOE must indicate or prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent malfunctions. Examples of environmental conditions are voltage, clock frequency, temperature, or external energy fields. Remark: A malfunction of the TOE may also be caused using a direct interaction with elements on the chip surface. This is considered as being a manipulation (refer to the objective O.Phys-Manipulation) provided that detailed knowledge about the TOE´s internal construction is required and the attack is performed in a controlled manner.

O. Phys-Manipulation: Protection against Physical Manipulation

The TOE must provide protection against manipulation of the TOE (including its software and Data), the Security IC Embedded Software and the User Data. This includes protection against

- reverse-engineering (understanding the design and its properties and functions),
- manipulation of the hardware and any data, as well as
- controlled manipulation of memory contents (Application Data).

The TOE must be designed and fabricated so that it requires a high combination of complex equipment, knowledge, skills, and time to be able to derive detailed design information or other information which could be used to compromise security through such a physical attack.

O. Leak-Forced: *Protection against Forced Information Leakage*

The Security IC must be protected against disclosure of confidential data processed in the Security IC (using methods as described under O.Leak-Inherent) even if the information leakage is not inherent but caused by the attacker

- by forcing a malfunction (refer to "Protection against Malfunction due to Environmental Stress (O.Malfunction)" and/or
- by a physical manipulation (refer to "Protection against Physical Manipulation (O.Phys-Manipulation)".

If this is not the case, signals which normally do not contain significant information about secrets could become an information channel for a leakage attack.

O. Abuse-Func: *Protection against Abuse of Functionality*

The TOE must prevent that functions of the TOE which may not be used after TOE Delivery can be abused in order to

- disclose critical User Data,
- manipulate critical User Data of the Security IC Embedded Software,
- bypass, deactivate, change or explore security features or security services of the TOE.

Details depend, for instance, on the capabilities of the Test Features provided by the IC Dedicated Software which are not specified here.

O. Identification: TOE Identification

The TOE must provide means to store Initialisation Data and Pre-personalisation Data in its non-volatile memory. The Initialisation Data (or parts of them) are used for TOE identification.

O.RND: Protection against Deficiency of Random Numbers

The TOE will ensure the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have a sufficient entropy.

The TOE will ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.

O.Add-Functions: Additional Specific Security Functions

The TOE must provide the following specific security functionality to the Smartcard Embedded Software:,

- Data Encryption Standard (DES)
- Triple Data Encryption Standard (3DES)
- Advanced Encryption Standard (AES256)
- Rivest-Shamir-Adleman (RSA1024, RSA2048)
- Cryptographic Key Generation for RSA2048 algorithm

4.2.Security Objectives for Development of Operating System

The development of the Security IC Embedded Software is outside the development and manufacturing of the TOE. The Security IC Embedded Software defines the operational use of the TOE. This section describes the security objectives for the operational environment enforced by the Security IC Embedded Software. The objectives for development of the Operating System are listed in Table 8 and explained below:

Table 8. Objectives for development of the Operating System

	OBJECTIVE	BRIEF DESCRIPTION
1.	OE.Plat-Appl	Usage of Hardware Platform
2.	OE.Resp-Appl	Treatment of User Data

Phase 1

OE.Plat-Appl: Usage of Hardware Platform

To ensure that the TOE is used in a secure manner the Security IC Embedded Software shall be designed so that the requirements from the following documents are met:

- UKTÜM-H v7.0 security requirements document
- Findings of the TOE evaluation reports relevant for the Security IC Embedded Software as referenced in the certification report.

Because the TOE implements additional specific security functionality (as in O.Add-Functions), OE.Plat-Appl covers the use of these functions by Smartcard Embedded Software as follows:

The TOE supports cipher schemes as additional specific security functionality. If required, the Smartcard Embedded Software shall use these cryptographic services of the TOE and their interface as specified. When key-dependent functions implemented in the Smartcard Embedded Software are just being executed, the Smartcard Embedded Software must provide protection against disclosure of confidential data (User Data) stored and/or processed in the

TOE by using the methods described under “Inherent Information Leakage (T.Leak-Inherent)” and “Forced Information Leakage (T.Leak-Forced)”.

OE.Resp-Appl: Treatment of User Data

Security relevant User Data (especially cryptographic keys) are treated by the Security IC Embedded Software as required by the security needs of the specific application context. For example the Security IC Embedded Software will not disclose security relevant User Data to unauthorised users or processes when communicating with a terminal.

Because the TOE implements additional specific security functionality (as inO.Add-Functions), OE.Resp-Appl covers the use of these functions by Smartcard Embedded Software as follows:

By definition cipher or plain text data and cryptographic keys are User Data. The Smartcard Embedded Software shall treat these data appropriately, use only proper secret keys (chosen from a large key space) as input for the cryptographic function of the TOE and use keys and functions appropriately in order to ensure the strength of cryptographic operation. This means that keys are treated as confidential as soon as they are generated. The quality and confidentiality of the keys must be maintained. This implies that appropriate key management has to be realised in the environment. The keys must be unique with a very rich probability, and cryptographically strong. For example, it must be ensured that it is beyond practicality to derive the private key from a public key if asymmetric algorithms are used.

4.3. Security Objectives for the Development Environment

The objectives for development environment are listed in Table 9 and explained below:

Table 9. Objectives for development environment

	OBJECTIVE	BRIEF DESCRIPTION
1.	OE. Process-TOE	Protection during TOE Development and Production
2.	OE. Process-SEC-IC	Protection during composite product manufacturing

Phases 2-4

OE. Process-TOE : Protection during TOE Development and Production

The TOE Manufacturer must ensure that the development and production of the Smartcard Integrated Circuit (Phases 2 - 4) is secure so that no information is unintentionally made available for the operational phase of the TOE. For example, the confidentiality and integrity of design information and test data must be guaranteed, access to samples, development tools and other material must be restricted to authorised persons only, scrap must be destroyed. This not only pertains to the TOE but also to all information and material exchanged with the developer of the Smartcard Embedded Software and therefore especially to the Smartcard Embedded Software itself. This includes the delivery (exchange) procedures for Phase 1 and the Phases after TOE Delivery.

An accurate identification must be established for the TOE. This requires that each instantiation of the TOE carries this unique identification. The accurate identification is introduced at the end of the production test in phase 3. Therefore the production environment must support this unique identification.

Phase 5-6

OE. Process-SEC-IC: Protection during composite product manufacturing

Security procedures shall be used after TOE Delivery up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use).

4.4. Security Objectives Rationale

Table 10 below gives an overview, how the assumptions, threats, and organisational security policies are addressed by the objectives. The text following after the table justifies this in detail.

Table 10. Coverage of Assumptions, Threats and Organisational Security Policies By Security Objectives

Assumptions, Threats or Policies	**Objectives**	**Rationale**
A.Key-Function	OE.Plat-Appl, OE.Resp-Appl	The Smartcard Embedded Software must implement functions which perform operations on keys in such a manner that they do not disclose information about confidential data. The non disclosure due to leakage A.Key-Function attacks is included in objective OE.Plat-Appl. By definition cipher or plain text data and cryptographic keys are User Data. So, the Smartcard Embedded Software will protect such data if required and use keys and functions appropriately in order to ensure the strength of cryptographic operation. Quality and confidentiality must be maintained for keys that are imported and/or derived from other keys. This implies that appropriate key management has to be realised in the environment. That is expressed by the assumption A.Key-Function which is covered from OE.Resp–Appl.
A.Plat-Appl	OE.Plat-Appl	Since OE.Plat-Appl requires the Security IC Embedded Software developer to implement those measures assumed in A.Plat-Appl, the assumption is covered by the objective.
A.Process-Sec-IC	OE.Process-Sec-IC	Since OE.Process-Sec-IC requires the Composite Product Manufacturer to implement those measures assumed in A.Process-Sec-IC, the assumption is covered by this objective.
A.Resp-Appl	OE.Resp-Appl	Since OE.Resp-Appl requires the developer of the Security IC Embedded Software to implement measures as assumed in A.Resp-Appl, the assumption is covered by the objective.
P.Add-Functions	O.Add-Functions	Since O.Add-Functions requires the TOE to implement exactly the same specific security functionality as required by P.Add-Functions,

		the organisational security policy is covered by the objective.
P.Process-TOE	OE.Process-TOE O.Identification	Unique identification is ensured by O.Identification and the development environment security is covered by OE.Process-TOE. Therefore OE.Process-TOE and O.Identification together covers P.Process-TOE
T.Leak-Inherent	O.Leak-Inherent	The objective O.Leak-Inherent is defined in a way that it directly corresponds to the description of the threat T.Leak-Inherent. It is clear from the description of the objective O.Leak-Inherent that the the threat T.Leak-Inherent is removed if the objective is valid.
T.Phys-Probing	O.Phys-Probing	The objective O.Phys-Probing is defined in a way that it directly corresponds to the description of the threat T.Phys-Probing. It is clear from the description of the objective O.Phys-Probing that the the threat T.Phys-Probing is removed if the objective is valid.
T.Phys-Manipulation	O.Phys-Manipulation	The objective O.Phys-Manipulation is defined in a way that it directly corresponds to the description of the threat T.Phys-Manipulation. It is clear from the description of the objective O.Phys-Manipulation that the the threat T.Phys-Manipulation is removed if the objective is valid.
T.Malfunction	O.Malfunction	The objective O.Malfunction is defined in a way that it directly corresponds to the description of the threat T.Malfunction It is clear from the description of the objective O.Malfunction that the the threat T.Malfunction is removed if the objective is valid.
T.Leak-Forced	O.Leak-Forced	The objective O.Leak-Forced is defined in a way that it directly corresponds to the description of the threat T.Leak-Forced. It is clear from the description of the objective O.Leak-Forced that the the threat T.Leak-Forced is removed if the objective is valid.
T.Abuse-Func	O.Abuse-Func	The objective O.Abuse-Func is defined in a way that it directly corresponds to the description of the threat T.Abuse-Func. It is clear from the description of the objective O.Abus-Func that the the threat T.Abus-Func is removed if the objective is valid.
T.RND	O.RND	The objective O.RND is defined in a way that it directly corresponds to the description of the threat T.RND. It is clear from the description of the objective O.RND that the the threat T.Leak-RND is removed if the objective is valid.

5. EXTENDED COMPONENTS DEFINITION

In this Security Target the same extended components as in PP are defined as below

5.1. Security Functional Component FPT_TST.2 TSF Self-testing

The following additions are made to “TSF self test (FPT_TST)” in Common Criteria, Part 2 to require the self-testing of TSF and of the integrity of the TSF-data and TSF-executable code. FPT_TST.2 requires the behaviour of TSF during self-testing and the actions to be performed by TSF in dependency of the results of self-testing. This kind of requirements lises beyond FPT_TST.1 defined in Common Criteria, Part 2.

-Component levelling : **FPT_TST.1** TSF Testing,

FPT_TST.2 TSF Self-Testing

FPT_TST.1 TSF testing provides the ability to test the TSF’s correct operation. These tests may be performed at start-up, periodically, at the request of the authorised user, or when other conditions are met. It also provides the ability to verify the integrity of TSF data and executable code.

FPT_TST.2 TSF self-tesing requires self-testing capabilities of TSF correct operation. These tests may be performed at start-up. Conditional and on demand by a user self-testing may be required. Particular TSF behaviour during self-testing and TSF-actions after self-testing are required.

The security functional component family “TSF Self-Testing (FPT_TST.2)” is specified as follows.

FPT_TST.2 TSF Self-Testing

Hierarchical to: No other components.

FPT_TST.2.1 The TSF shall run a suite of self tests [*selection: during initial startup, periodically during normal operation, at the request of the authorised user, and/or at the conditions* [assignment: conditions under which test should occur]] to demonstrate the correct operation of [assignment:functions and/or mechanisms].

Dependencies: FPT_AMT.1 Abstract machine testing

5.2.FMT_LIM Limited Capabilities and Availability

Definition of the Family

To define the IT security functional requirements of the TOE an additional family (FMT_LIM) of the Class FMT (Security Management) is defined here. This family describes the functional requirements for the Test Features of the TOE. The new functional requirements were defined in the class FMT because this class addresses the management of functions of the TSF. The examples of the technical mechanism used in the TOE appropriate to address the specific issues of preventing the abuse of functions by limiting the capabilities of the functions and by limiting their availability.

The family "Limited capabilities and availability (FMT_LIM)" is specified as follows.

Family behaviour

This family defines requirements that limit the capabilities and availability of functions in a combined manner. Note that FDP_ACF restricts the access to functions whereas the component Limited capability of this family requires the functions themselves to be designed in a specific manner.

Component Leveling

FMT_LIM.1 Limited capabilities requires that the TSF is built to provide only the capabilities (perform action, gather information) necessary for its genuine purpose.

FMT_LIM.2 Limited availability requires that the TSF restrict the use of functions (refer to Limited capabilities (FMT_LIM.1)). This can be achieved, for instance, by removing or by disabling functions in a specific phase of the TOE's life-cycle.

Management: FMT_LIM.1, FMT_LIM.2

There are no management activities foreseen.

Audit: FMT_LIM.1, FMT_LIM.2

Components:

The TOE Functional Requirement "Limited capabilities (FMT_LIM.1)" is specified as follows.

FMT_LIM.1 Limited Capabilities

Hierarchical to: No other components.

FMT_LIM.1.1 The TSF shall be designed and implemented in a manner that limits its capabilities so that in conjunction with “Limited availability (FMT_LIM.2)” the following policy is enforced [assignment: Limited capability and availability policy].

Dependencies: FMT_LIM.2 Limited availability.

FMT_LIM.2 Limited Availability

Hierarchical to: No other components.

FMT_LIM.2.1 The TSF shall be designed and implemented in a manner that limits its availabilities so that in conjunction with “Limited capability (FMT_LIM.1)” the following policy is enforced [assignment: Limited capability and availability policy].

Dependencies: FMT_LIM.1 Limited capabilities.

Application note: The functional requirements FMT_LIM.1 and FMT_LIM.2 assume that there are two types of mechanisms (limitation of capabilities and limitation of availability) which together shall provide protection in order to enforce the policy. This also allows that

i. the TSF is provided without restrictions in the product in its user environment but its capabilities are so limited that the policy is enforced

or conversely,

ii. the TSF is designed with high functionality but is removed or disabled in the product in its user environment.

The combination of both requirements shall enforce the policy.

5.3.FAU_SAS Audit Data Storage

Definition of the Family

To define the security functional requirements of the TOE an additional family (FAU_SAS) of the Class FAU (Security Audit) is defined here. This family describes the functional requirements for the storage of audit data. It has a more general approach than FAU_GEN, because it does not necessarily require the data to be generated by the TOE itself and because it does not give specific details of the content of the audit records.

The family "Audit data storage (FAU_SAS)" is specified as follows.

Family behaviour

This family defines functional requirements for the storage of audit data.

Component Leveling

FAU_SAS.1	Requires the TOE to provide the possibility to store audit data.

Management: **FAU_SAS.1**

There are no management activities foreseen.

Audit: **FAU_SAS.1**

There are no actions defined to be auditable.

Components:

FAU_SAS.1 Audit Storage

Hierarchical to: No other components.

FAU_SAS.1.1 The TSF shall provide [assignment: list of subjects] with the capability to store [assignment: list of audit information] in the [assignment: type of persistent memory].

Dependencies: No dependencies

5.4.FCS_RND Generation of random numbers

Definition of the Family FCS_RND

To define the IT security functional requirements of the TOE an additional family (FCS_RND) of the Class FCS (cryptographic support) is defined here. This family describes the functional requirements for random number generation used for cryptographic purposes.

Family behaviour

This family defines quality requirements for the generation of random numbers which are intended to be used for cryptographic purposes.

Component levelling:

FCS_RND.1 Generation of random numbers requires that random numbers meet a defined quality metric.

Management: FCS_RND.1

There are no management activities foreseen.

Audit:

There are no actions defined to be auditable.

Components:

FCS_RND.1 Generation of random numbers

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RND.1.1 The TSF shall provide a [*selection: physical, non-physical true, deterministic, hybrid*] random number generator that implements: [assignment: list of security capabilities].

FCS_RND.1.2 The TSF shall provide random numbers that meet [assignment: a defined quality metric].

Application Note : A physical random number generator (RND) produces the random number by a noise source based on physical random processes. A non-physical true RNG uses a noise source based on non-physical random processes like human interaction (key strokes, Mouse movement). A deterministic RNG uses a random seed to produce a pseudorandom output. A hybrid RNG combines the principles of physical and deterministic RNGs.

6. IT SECURITY REQUIREMENTS

6.1. Security Functional Requirements for the TOE

Security functional requirements for the TOE are grouped and listed below. They are also given in Table 11.

Standard SFRs Protecting User Data and Also Supporting the Other SFRs :

Malfunctions:

-FRU_FLT.2 : Limited Fault Tolerance

-FPT_FLS.1 : Failure With Preservation of Secure State

-FPT_TST.2 : Subset TOE Security Testing

-FDP_SDI.2: Stored Data Integrity Monitoring and Action

Leakage:

-FDP_ITT.1 : Basic Internal Transfer Protection

-FPT_ITT.1 : Basic Internal TSF Data Transfer Protection

-FPT_PHP.3 : Resistance to Physical Attack

-FDP_IFC.1 : Subset Information Flow Control

Standard SFRs Supporting TOE's Life Cycle and Preventing Abuse of Functions :

Abuse of Functionality:

-FMT_LIM.1 : Limited Capabilities

-FMT_LIM.2 : Limited Availability

Identification:

-FAU_SAS.1 : Audit Storage

SFRs Related To Special Functionality :

Random Numbers

-FCS_RND.1 : Random Numbers

Cryptographic Operations

-FCS_COP.1 : Cryptographic Operation

-FCS_CKM.1 : Cryptographic Key Management

Table 11. Security Functional Requirements for the TOE

Security Class	SFR		Refinement	Covered Objectives
FRU : Resource Utilization	FRU_FLT.2	Limited Fault Tolerance	Yes	O.Malfunction, O.Abuse-Func, O.Leak-Forced, O.RND
FPT: Protection of the TSF	FPT_FLS.1	Failure with preservation of secure state	Yes	O.Malfunction, O.Abuse-Func, O.Leak-Forced, O.RND
	FPT_PHP.3	Resistance to Physical Attack	Yes	O.Phys-Probing O.Phys-Manipulation O.Leak-Forced O.Abuse-Func O.RND
	FPT_ITT.1	Basic internal TSF data transfer protection	Yes	O.Leak-Inherent, O.Leak-Forced, O.Abuse-Func, O.RND
	FPT_TST.2	TSF Testing	No	O.Phys-Manipulation

FDP: USER Data Protection	FDP_ITT.1	Basic Internal Transfer Protection	Yes	O.Leak-Inherent, O.Leak-Forced, O.Abuse-Func, O.RND
	FDP_IFC.1	Subset Information Flow Control	Yes	O.Leak-Inherent, O.Leak-Forced, O.Abuse-Func, O.RND
	FDP_SDI.2	Stored Data Integrity Monitoring and Action	No	O.Malfunction
FMT: Security Management	FMT_LIM.1	Limited Capabilities	Yes	O.Abuse-Func
	FMT_LIM.2	Limited Availability	No	O.Abuse-Func
FAU: Security Audit	FAU_SAS.1	Audit Storage	No	O.Identification, OE.Process-TOE
FCS: Cryptographic Support	FCS_RND.1	Random Number Generation	No	O.RND
	FCS_COP.1 – iteration-1 (DES)	Cryptographic Operation	No	O.Add-Functions

	FCS_COP.1 – iteration-2 (3DES)	Cryptographic Operation	No	O.Add-Functions
	FCS_COP.1 – iteration-3 (AES)	Cryptographic Operation	No	O.Add-Functions
	FCS_COP.1 – iteration-4 (RSA)	Cryptographic Operation	No	O.Add-Functions
	FCS_CKM.1 – RSA	Cryptographic Key Generation	No	O.Add-Functions

6.1.1.FRU Resource Utilization

6.1.1.1. FRU_FLT Fault Tolerance

FRU_FLT.2 Limited Fault Tolerance

FRU_FLT.2.1 The TSF shall ensure the operation of all the TOE’s capabilities when the following failures occur: [exposure to operating conditions which are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1)]

Refinement : The term “failure” above means “circumstances”. The TOE prevents failures for the “circumstances” defined above. The TOE operates without failure when the sensors do not raise the ALARM signal.

6.1.2.FPT Protection of the TSF

6.1.2.1. FPT_FLS Fail Secure

FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur:

[a) The temperature of the operating environment goes out of the specified ranges;

b) The external supply voltage goes out of the specified ranges;

c) Clock frequency goes out of the specified ranges].

Refinement: <u>The term “failure” above also covers “circumstances”. The TOE prevents failures for the circumstances” defined above. When the sensors which senses these conditions raises the ALARM signal, the device enters to reset state</u>.

6.1.2.2. FPT_PHP TSF Physical Protection

FPT_PHP.3 Resistance to Physical Attack

FPT_PHP.3.1 The TSF shall resist [physical manipulation and physical probing] to the [microprocessor, ciphering blocks, SRAM and Flash memories, data and address busses between microprocessor and ciphering blocks, data busses between microprocessor and SRAM and Flash memories storing the user data, data and address busses between microprocessor and Flash memories storing the Security IC Embedded Software] by responding automatically such that the SFRs are always enforced.

Refinement : The TSF will implement appropriate mechanisms to continuously counter physical manipulation and physical probing. Due to the nature of these attacks (especially manipulation) the TSF can by no means detect attacks on all of its elements. Therefore, permanent protection against these attacks is required ensuring that security functional requirements are enforced. Hence, "automatic response" means here (i) assuming that there might be an attack at any time and (ii) countermeasures are provided at any time.

6.1.2.3. FPT_ITT Internal TOE TSF Data

FPT_ITT.1 Basic internal TSF data transfer protection

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

Refinement: The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as separated parts of the TOE.

Refinement: This requirement is equivalent to FDP_ITT.1 above but refers to TSF data instead of User Data. Therefore, it should be understood as to refer to the same Data Processing Policy defined under FDP_IFC.1 below.

6.1.2.4. FPT_TST TSF Self Test

FPT_TST.2 Subset TOE Security Testing

FPT_TST.2.1 The TSF shall run a suite of self tests [*during initial startup* and at the cases that the operating system requires] to demonstrate the correct operation of [active shield, security sensors, random number generator and DES-3DES, AES, RSA cryptographic operations].

6.1.3.FDP User Data Protection

6.1.3.1. FDP_ITT Internal TOE Transfer

FDP_ITT.1 Basic Internal Transfer Protection

FDP_ITT.1.1 The TSF shall enforce the [Data Processing Policy defined under FDP_IFC.1] to prevent the [*disclosure*] of user data when it is transmitted between physically-separated parts of the TOE.

Refinement : The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as separated parts of the TOE.

6.1.3.2. FDP_IFC Information Flow Control Policy

FDP_ IFC.1 Subset Information Flow Control

FDP_IFC.1.1 The TSF shall enforce [the Data Processing Policy] on [all confidential data when they are processed or transferred by the TOE or by the Security IC Embedded Software].

Refinement : Data Processing Policy : User Data and TSF data shall not be accessible from the TOE except when the Security IC Embedded Software decides to communicate the User Data via an external interface. The protection shall be applied to confidential data only but without the distinction of attributes controlled by the Security IC Embedded Software.

6.1.3.3. FDP_SDI Stored Data Integrity

FDP_SDI.2 Stored Data Integrity Monitoring and Action

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for [*unmatched contents with related checksums*] on all objects, based on the following attributes: [*checkbits for Internal RAM and SRAM*].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall [*produce an internal reset related to data reads with unmatched checkbits in Internal RAM and SRAM*].

Refinement : In addition to the TOE's monitoring and action capabilities on Internal RAM and SRAM, the Smartcard Embedded Software may contain its original algoritm or may use the CRC hardware of the TOE to calculate its own checksum and the partial checksums of the data stored in the Flash memory. The Smartcard Embedded Software may generate error messages when calculated checksum values does not match with the stored values.

6.1.4.FCS Cryptographic Support

6.1.4.1. FCS_COP Cryptographic Operation

FCS_COP.1 Cryptographic Operation: Iteration [1], DES

FCS_COP.1.1 The TSF shall perform [encryption and decryption operations] in accordance with a specified cryptographic algorithm [Data Encryption Standard (DES)] and cryptographic key sizes [56 bit] that meet the following:

[U.S Department of Commerce / National Bureau of Standards Data Encryption Standard (DES), FIPS PUB 46-3, 1999 October 25.]

FCS_COP.1 Cryptographic Operation: Iteration [2], TRIPLE DES

FCS_COP.1.1 The TSF shall perform [encryption and decryption operations] in accordance with a specified cryptographic algorithm [Triple Data Encryption Standard (3DES)] and cryptographic key sizes [112 bit] that meet the following:

[U.S Department of Commerce / National Bureau of Standards Data Encryption Standard (DES), FIPS PUB 46-3, 1999 October 25. keying option 2]

FCS_COP.1 Cryptographic Operation: Iteration [3], AES Operation

FCS_COP.1.1 The TSF shall perform [encryption and decryption operations] in accordance with a specified cryptographic algorithm [Advanced Encryption Standard (AES)] and cryptographic key sizes [256 bit] that meet the following:

[U.S Department of Commerce / National Institute of Standards and Technology Advanced Encryption Standard (AES), FIPS PUB 197, 2001 November 26.]

FCS_COP.1 Cryptographic Operation: Iteration [4], RSA Operation

FCS_COP.1.1 The TSF shall perform [encryption and decryption operations] in accordance with a specified cryptographic algorithm [Rivest-Shamir-Adleman (RSA)] and cryptographic key sizes [1024 bit and 2048 bit] that meet the following:

[ISO/IEC 9796-1, Annex A, sections A.4 and A.5, and Annex C.]

6.1.4.2. FCS_CKM Cryptographic key management

FCS_CKM.1 Cryptographic Operation: RSA Key Generation

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [stated as in ‘TCKK Akıllı Kartlarında RSA İmzalama Anahtar Çifti Üretimi’, TKRD-501-11-TA-TR01, 23 Ocak 2012,BİLGEM-UEKAE] and specified cryptographic key sizes of [2048 bit] that meet the following: [FIPS 186-3, 2009 June].

6.1.4.3. FCS_RND Random Number Generation

FCS_RND Random Number Generation

FCS_RND.1.1 The TSF shall provide a [*physical*] random number generator that implements [total failure test of the random source].

FCS_RND.1.2 The TSF shall provide random numbers that meet [the requirements of monobit, poker, runs, long run, and auto correlation tests defined in FIBS-140-1 and pass all these tests for 20.000 bit length].

6.1.5.FMT Security Management

6.1.5.1. FMT_LIM Limited Capabilities and Availability

FMT_LIM.1 Limited Capabilities

FMT_LIM.1.1 The TSF shall be designed and implemented in a manner that limits its capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced, [after the submission of the TOE (After Phase 4), Self Test Software which is part of the IC Dedicated Software does not permit to collect any data which causes to disclose or change the User Data and/or the TSF data or any other].

Refinement: <u>'Capabilities' are the functions implemented in the Self Test Software</u>.

FMT_LIM.2 Limited Availability

FMT_LIM.2.1 The TSF shall be designed and implemented in a manner that limits its availability so that in conjunction with "Limited capability (FMT_LIM.1)" the following policy is enforced, [after the submission of the TOE (After Phase 4), Self Test Software which is part of the IC Dedicated Software does not permit to collect any data which causes to disclose or change the User Data and/or the TSF data or any other].

Refinement: <u>'Availability' is availability of the functions implemented in the Self Test Software</u> .

6.1.6.FAU Security Audit

6.1.6.1. FAU_SAS Audit Data Storage

FAU_SAS.1 Audit Storage

FAU_SAS.1.1	The TSF shall provide [test process] with the capability to store [initialisation data and/or pre-personalisation data before the TOE submission (Before Phase 5)] in the [Flash memory].

6.2. Security Assurance Requirements of the TOE

The assurance level of this security target document is EAL 5+ (AVA_VAN.5). The assurance components of this package is given in Table 12.

Table 12. TOE Assurance Components

Assurance Class	Component ID	Component Title
Development	ADV_ARC.1	Security architecture description
	ADV_FSP.5	Complete Semi-Formal functional specification with additional error information
	ADV_IMP.1	Implementation representation of the TSF
	ADV_INT.2	Well-Structured Internals
	ADV_TDS.4	Semiformal modular design
Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-cycle support	ALC_CMC.4	Production support, acceptance procedures and automation
	ALC_CMS.5	Development Tools CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_DVS.1	Identification of security measures
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.2	Compliance With Implementation Standards
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
Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.3	Testing: modular design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
Vulnerability assessment	AVA_VAN.5	Advanced Methodological Vulnerability Analysis

6.3. Security Functional Requirements Rationale

Security Functional Requirements Rationale is given in Table 13. For all the objectives below, additional required support by the Security IC Embedded Software are addressed in the UKTÜM-H v7.0 Security Requirement for the Operating System document.

Table 13. Coverage of Objectives by Security Functional R equirements

Target	SFR	Rationale
O.Leak-Inherent	FDP_ITT.1 FPT_ITT.1 FDP_IFC.1	The refinements of the security functional requirements FPT_ITT.1 and FDP_ITT.1 together with the policy statement in FDP_IFC.1 explicitly require the prevention of disclosure of secret data (TSF data as well as User Data) when transmitted between separate parts of the TOE or while being processed. This includes that attackers cannot reveal such data by measurements of emanations, power consumption or other behaviour of the TOE while data are transmitted between or processed by TOE parts. Together with this FPT_ITT.1, FDP_ITT.1 and FDP_IFC.1 are suitable to meet the objective.
O.Phys-Probing	FPT_PHP.3	The scenario of physical probing as described for this objective is explicitly included in the assignment chosen for the physical tampering scenarios in FPT_PHP.3. Therefore, it is clear that this security functional requirement supports the objective. Together with this FPT_PHP.3 is suitable to meet the objective.
O.Phys-Manipulation	FPT_PHP.3 FPT_TST.2	The scenario of physical manipulation as described for this objective is explicitly included in the assignment chosen for the physical tampering scenarios in FPT_PHP.3. Therefore, it is clear that this security functional requirement supports the objective. Since the security functional requirement FPT.TST.2 will detect any attempt to conduce a physical manipulation aiming to overcome security enforcing functions, it supports the objective.
O.Malfunction	FRU_FLT.2 FPT_FLS.1 FDP_SDI.2	The definition of this objective shows that it covers a situation, where malfunction of the TOE might be caused by the operating conditions of the TOE (while direct manipulation of the TOE is covered O.Phys-Manipulation). There are two possibilities in this situation: Either the operating conditions are inside the tolerated range or at least one of them is outside of this range. The second case is covered by FPT_FLS.1, because it states that a secure state is preserved in this case. The first case is covered by FRU_FLT.2 because it

		states that the TOE operates correctly under normal (tolerated) conditions. The functions implementing FRU_FLT.2 and FPT_FLS.1 must work independently so that their operation can not affected by the Security IC Embedded Software (refer to the refinement). Therefore, there is no possible instance of conditions under O.Malfunction, which is not covered. The TOE will enter to reset state for any data read from internal RAM or SRAM with unmatched check bits satisfying FDP_SDI.2.
O.Leak-Forced	FDP_ITT.1 FPT_ITT.1 FDP_IFC.1 FRU_FLT.2 FPT_FLS.1 FPT_PHP.3	This objective is directed against attacks, where an attacker wants to force an information leakage, which would not occur under normal conditions. In order to achieve this the attacker has to combine a first attack step, which modifies the behaviour of the TOE (either by exposing it to extreme operating conditions or by directly manipulating it) with a second attack step measuring and analysing some output produced by the TOE. The first step is prevented by the same mechanisms which support O.Malfunction and O.Phys-Manipulation, respectively. The requirements covering O.Leak-Inherent also support O.Leak-Forced because they prevent the attacker from being successful if he tries the second step directly.
O.Abuse-Func	FMT_LIM.1 FMT_LIM.2 FDP_ITT.1 FPT_ITT.1 FDP_IFC.1 FPT_PHP.3 FRU_FLT.2 FPT_FLS.1	This objective states that abuse of functions (especially provided by the IC Dedicated Software, for instance in order to read secret data) must not be possible in Phase 7 of the life-cycle. There are two possibilities to achieve this: (i) They cannot be used by an attacker (i. e. its availability is limited) or (ii) using them would not be of relevant use for an attacker (i. e. its capabilities are limited) since the functions are designed in a specific way. The first possibility is specified by FMT_LIM.2 and the second one by FMT_LIM.1. Since these requirements are combined to support the policy, which is suitable to fulfill O.Abuse-Func, both security functional requirements together are suitable to meet the objective. Other security functional requirements which prevent attackers from circumventing the functions implementing these two security functional requirements (for instance by manipulating the hardware) also support the objective. It was chosen to define FMT_LIM.1 and FMT_LIM.2 explicitly (not using Part 2 of the Common Criteria) for the following reason: Though taking components from the Common Criteria catalogue makes it easier to recognise functions, any selection from Part 2 of the Common Criteria would have made it harder for the reader to understand the special situation meant here. As a consequence, the statement of explicit security functional requirements was chosen to provide more clarity.

O.Identification	FAU_SAS.1	Obviously the operations for FAU_SAS.1 are chosen in a way that they require the TOE to provide the functionality needed for O.Identification. The Initialisation Data (or parts of them) are used for TOE identification. The technical capability of the TOE to store Initialisation Data and/or Pre-personalisation Data is provided according to FAU_SAS.1. It was chosen to define FAU_SAS.1 explicitly (not using a given security functional requirement from Part 2 of the Common Criteria) for the following reason: The security functional requirement FAU_GEN.1 in Part 2 of the CC requires the TOE to generate the audit data and gives details on the content of the audit records (for instance data and time). The possibility to use the functions in order to store security relevant data which are generated outside of the TOE, is not covered by the family FAU_GEN or by other families in Part 2. Moreover, the TOE cannot add time information to the records, because it has no real time clock. Therefore, the new family FAU_SAS was defined for this situation.
O.RND	FCS_RND.1 FDP_ITT.1 FPT_ITT.1 FDP_IFC.1 FPT_PHP.3 FRU_FLT.2 FPT_FLS.1 FPT_TST.2	FCS_RND.1 requires the TOE to provide random numbers of good quality. Other security functional requirements, which prevent physical manipulation and malfunction of the TOE (see the corresponding objectives listed in the table) support this objective because they prevent attackers from manipulating or otherwise affecting the random number generator. Random numbers are often used by the Security IC Embedded Software to generate cryptographic keys for internal use. Therefore, the TOE must prevent the unauthorised disclosure of random numbers. Other security functional requirements which prevent inherent leakage attacks, probing and forced leakage attacks ensure the confidentiality of the random numbers provided by the TOE. Depending on the functionality of specific TOEs the Security IC Embedded Software will have to support the objective by providing runtime-tests of the random number generator. Together, these requirements allow the TOE to provide cryptographically good random numbers and to ensure that no information about the produced random numbers is available to an attacker. It was chosen to define FCS_RND.1 explicitly, because Part 2 of the Common Criteria do not contain generic security functional requirements for Random Number generation. (Note, that there are security functional requirements in Part 2 of the Common Criteria, which refer to random numbers. However, they define requirements only for the authentication context, which is only one of the possible applications of random numbers.)
O.Add-Functions	FCS_COP.1 –	Since O.Add-Functions requires the TOE to implement

	iteration[1] FCS_COP.1 – iteration[2] FCS_COP.1 – iteration[3] FCS_COP.1 – iteration[4] FCS_CKM.1 – RSA	exactly the same specific security functionality as required by P.Add-Functions; the organisational security policy is covered by the objective. Nevertheless the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced define how to implement the specific security functionality required by P.Add-Functions. (Note that these objectives support that the specific security functionality is provided in a secure way as expected from P.Add-Functions.) Especially O.Leak- Inherent and O.Leak-Forced refer to the protection of confidential data (User Data or TSF data) in general. User Data are also processed by the specific security functionality required by P.Add-Functions.

Dependencies of Security Functions are given in Table 14:

Table 14. Dependencies of Security Functional Requirements

Component	**Dependencies**	**Satisfied by security requirements in this ST?**
FMT_LIM.1	FMT.LIM.2	Yes
FMT_LIM.2	FMT.LIM.1	Yes
FAU_SAS.1	None	No dependency
FCS_RND.1	None	No dependency
FPT_TST.2	FPT_AMT.1	See discussion below
FRU_FLT.2	FPT_FLS.1	Yes
FPT_FLS.1	None	No dependency
FPT_PHP.3	None	No dependency
FPT_ITT.1	None	No dependency
FDP_ITT.1	FDP_IFC.1	Yes
FDP_IFC.1	FDP_IFF.1	See discussion below

FDP_SDI.2	None	No dependency
FCS_COP.1	FDP_ITC.1 FDP_ITC.2 FCS_CKM.1 FCS_CKM.4 FMT_MSA.2	See discussion below
FCS_CKM.1/RSA	FCS_CKM.2 FCS_COP.1 FCS_CKM.4	See discussion below

The following discussion demonstrates how the dependencies defined by Part 2 of the Common Criteria for the requirement FDP_IFC.1 are satisfied:

Part 2 of the Common Criteria defines the dependency of FDP_IFC.1 (information flow control policy statement) on FDP_IFF.1 (Simple security attributes). The specification of FDP_IFF.1 would not capture the nature of the security functional requirement nor add any detail. As stated in the Data Processing Policy referred to in FDP_IFC.1 there are no attributes necessary. The security functional requirement for the TOE is sufficiently described using FDP_ITT.1 and its Data Processing Policy (FDP_IFC.1).

The following discussion demonstrates how the dependencies defined by Part 2 of the Common Criteria for the requirement FPT_TST.2 are satisfied:

The dependency defined in the Common Criteria is Abstract machine testing (FPT_AMT.1). Description of FPT_AMT.1 can be found in Protection Profile which is used as guidance for this ST. Part 2 of the Common Criteria explains that the term “underlying abstract machine” typically refers to the hardware components upon which the TSF has been implemented. However, the phrase can also be used to refer to an underlying, previously evaluated hardware and software combination behaving as a virtual machine upon which the TSF relies. “The TOE is already a platform representing the lowest level in a Smartcard. There is no lower or “underlying abstract machine” used by the TOE which can be tested. There is no need to perform testing according to FPT_AMT.1 and the dependency in the requirement FPT_TST.2 is therefore considered to be satisfied.

The following discussion demonstrates how the dependencies defined by Part 2 of the Common Criteria for the requirements FCS_COP.1 and FCS_CKM.1/RSA are satisfied:

These requirements all address the appropriate management of cryptographic keys used by the specified cryptographic function and are not part of the PP. Most requirements concerning key management shall be fulfilled by the environment since the Smartcard Embedded Software is designed for a specific application context and uses the cryptographic functions provided by the TOE. For the security functional requirement FCS_COP.1/DES and FCS_COP.1/AES the respective dependencies FCS_CKM.1, FCS_CKM.4 and FDP_ITC.1 or FDP_ITC.2 have to be fulfilled by the environment. This means that the environment shall meet the requirements FCS_CKM.1 and FCS_CKM.4 and shall meet the requirements FDP_ITC.1 or FDP_ITC.2. For the security functional requirement FCS_COP.1/RSA, the respective dependencies FCS_CKM.4 and FDP_ITC.1 or FDP_ITC.2 have to be fulfilled by the environment. This means that the environment shall meet the requirements FDP_ITC.1 or FDP_ITC.2. The respective dependency FCS_CKM.1 has to be fulfilled by the TOE with the security functional requirement FCS_CKM.1/RSA (for FCS_COP.1/RSA). Additionally the requirement FCS_CKM.1 can be fulfilled by the environment. For the security functional requirement FCS_CKM.1/RSA the respective dependency FCS_COP.1 is fulfilled by the TOE. The environment covers the respective dependency FCS_CKM.4. This means that the environment shall meet the requirement FCS_CKM.4.

Security Assurance Rationale is given in Table 15:

Table 15. Security Assurance Rationale

Related Assurance Family	Assurance Component	Satisfied By
Development (ADV)	ADV_ARC.1	Security Architecture Document
	ADV_FSP.5	Functional Specification Document
	ADV_IMP.1	Source Code
	ADV_INT.2	Internal Document
	ADV_TDS.4	TOE Design Document
Guidance (AGD)	AGD_OPE.1	User Guidance Document

	AGD_PRE.1	Preparative Procedures Document
Life Cycle Support(ALC)	ALC_CMC.4	Configuration Management Document
	ALC_CMS.5	Configuration Management Document
	ALC_DEL.1	Delivery Document
	ALC_DVS.1	Life Cycle Document
	ALC_LCD.1	Life Cycle Document
	ALC_TAT.2	Tools And Techniques Document
Security Target (ASE)	ASE_CCL.1	Security Target Document
	ASE_ECD.1	Security Target Document
	ASE_INT.1	Security Target Document
	ASE_OBJ.2	Security Target Document
	ASE_REQ.2	Security Target Document
	ASE_SPD.1	Security Target Document
	ASE_TSS.1	Security Target Document
Tests (ATE)	ATE_COV.2	Test Document
	ATE_DPT.3	Test Document
	ATE_FUN.1	Test Document
	ATE_IND.2	Test Document
Vulnerability Analysis (AVA)	AVA_VAN.5	Vulnerability Analysis Tests

7. TOE SUMMARY SPECIFICATION

7.1. TOE Security Functions

The TOE includes the following 7 security functions that meet security functional requirements mentioned in this ST:

SEF1: Guarantee of Correct Operation

SEF2: Phase Management

SEF3: Physical Protection Against Physical Probing and Manipulation

SEF4: Logical Protection Againts Data Leakage

SEF5: Random Number Generation

SEF6: TSF Self Test

SEF7: Cryptographic Support

The following description of the security enforcing functions is a complete representation of the TSF.

7.1.1. SEF1: Guarantee of Correct Operation

The TOE which can only be operated correctly under the specified conditions is equipped with different type of sensors monitoring the operating parameters to detect if the specified operating conditions are fulfiled. For this purpose, TOE includes temperature sensors, supply voltage sensor and external clock frequency sensor. If one of these sensors raises an alarm due to a violation in the operating conditions, than the circuit enters to reset state. Exposing the TOE to extreme operating conditions may be used by an attacker to modify TOE's behaviour to force information leakage or to affect the TOE to produce cryptographically bad random numbers. The presence of security sensors prevents such attacks.

These functions satisfy FPT_FLS.1 "Failure with Preservation of Secure State" requirement.

On the other hand, when these sensors do not raise any alarm, the TOE functions properly, thus, FRU_FLT.2 "Limited Fault Tolerance" requirement is satisfied.

7.1.2. SEF2: Phase Management

During the chip development and production phases of the life cycle (Phase 2,3,4), the TOE is always in Test Mode enabling the operation of the IC Dedicated Software which is used to perform the die tests and to inject pre-personalisation data to the correctly working chips. After TOE delivery (Phase 5-7), the TOE is in User Mode where IC Dedicated Software is irreversibly disabled and the operation of the Smartcard Embedded Software is made available.

During start-up of the circuit, IC Dedicated Software decides whether it is in the User Mode or the Test Mode by checking some phase management flags. If it is in Test Mode, the TOE requests authentication before doing any other operation. Thus FMT.LIM.1 and FMT.LIM.2 requirements are satisfied.

Both in Test Mode and User Mode, the chip identification data and pre-personalisation data can be accessible satisfying FAU_SAS.1

7.1.3. SEF3: Physical Protection Againts Physical Probing and Manipulation

There exist different measures to protect the design of the TOE and the user data stored in the TOE when the TOE is in operation and also when the power is not applied to the TOE.

An active shield formed by the metal lines with active signals protects the entire surface of the TOE against physical attacks. Since physical attacks over the surface need to modify the active shield lines, the detection of opened or shortened lines will notify a physical attack causing the circuit to enter to reset state.

The entire surface of the TOE is covered by metal lines with active signals in order to prevent the attacker from probing and acquiring any useful data.

The layout of the logic circuit including the microprocessor core is effectively randomised making it difficult to determine specific functional areas for reverse engineering.

The microprocessor in UKTÜM-H v7.0 is designed in a unique and non standard way. Therefore, reverse engineering works need much more effort.

In the TOE, the data and address busses between microprocessor and the DES, the AES and the RSA blocks are encrypted against probing.

In the TOE, the data is encrypted in the SRAM and in the Flash memory. Thus, there are no plain data on the busses between microprocessor and memories.

In the TOE, the data and address busses are encrypted in the NVM where the operating system is embedded. Thus, the data and address busses are encrypted between NVM and the microprocessor.

Even if the attacker reads the content of the NVM by reverse engineering, since the data is encrypted, the attacker does not obtain any useful data about the microprocessors software.

The internal memory of the microprocessor and the external SRAM block are equipped with check bits to prevent the attacks aiming to modify memory contents.

The critical registers of the TOE are dual implemented againts manipulation attacks When a manipulation attack is detected, the chip enters reset state.

All these measures prevent forced information leakage which may be realised by physically manipulating and probing the critical lines of the TOE . The confidentiality of the random numbers provided by the TOE is also ensured by these measures preventing probing and manipulation.

These measures satisfy the security functional requirement of FPT_PHP.3, “Resistance to physical attack”.

The TOE will enter to reset state for any data read from internal RAM or SRAM with unmatched check bits satisfying FDP_SDI.2. The TOE is equipped with CRC hardware that the Security IC Embedded Software may use to calculate the total or partial cheksums of ROM and Flash Memory to control the stored data.

7.1.4. SEF4: Logical Protection Againts Data Leakage

In order to protect TOE against data analysis on stored and internally transferred data, the data is encrypted on chip before it is written in the SRAM and flash memories.

The use of encryption in the communication between the DES, the AES, the RSA blocks and the microprocessor prevents the interpretation of the leaked data. Random data is inserted into the data and address busses on the same purpose.

All these measures are implemented to prevent an attacker to be successful by measuring and analysing emanations, power consumption and other outputs produced by the TOE.

The hardware implementation of the DES, the AES and the RSA algorithms are implemented to be resistant against side channel attacks. This prevents the secure data leakage.

These security functions of the TOE cover the FDP_ITT.1 “Basic Internal Transfer Protection” and FTP_ITT.1 “Basic Internal TSF Data Transfer Protection”. The encryption covers the “Data Processing Policy” and FDP_IFC.1 “Subset Information Flow Control”.

7.1.5. SEF5: Random Number Generation

The UKTÜM-H v7.0 is equipped with a physical random number generator which generates truly random numbers. The generated random numbers can be used by the operating system software and also by TOE’s security enforcing functions. The TOE has the capability to subject the generated numbers to the monobit, poker, runs, long run and auto correlation tests defined in FIBS-140-2. The covered security functional requirement is FCS_RND.1.

7.1.6. SEF6: TSF Self Test

The TOE has the hardware supports making available the test of its security enforcing functions SEF1, SEF5, SEF7 and partially SEF3 by the operating system software. Since TSF self test will detect the attempts to modify sensor devices, random number generator, active shield and DES-3DES, AES, RSA cryptographic operations the covered security functional requirement is FPT_TST.2.

7.1.7. SEF7: Cryptographic Support

The TOE is equipped with the hardware implementations of the DES-3DES, AES256, RSA1024 and RSA2048 cryptographic functions. The covered security functional requirement is FCS_COP.1.

The TOE shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [stated as in 'TCKK Akıllı Kartlarında RSA İmzalama Anahtar Çifti Üretimi', TKRD-501-11-TA-TR01, 23 Ocak 2012,BİLGEM-UEKAE] and specified cryptographic key sizes of [2048 bit] that meet [FIPS 186-3, 2009 June]. The covered security functional requirement is FCS_CKM.1/RSA.

7.2.TOE Security Functions Rationale

How the above mentioned security functions of the TOE meets the Security Functional Requirements are given in Table 16.

Table 16. Coverage of Security Functions Rationale

	SEF1	SEF2	SEF3	SEF4	SEF5	SEF6	SEF7
FRU_FLT.2	X						
FPT_FLS.1	X						
FMT_LIM.1		X					
FMT_LIM.2		X					
FAU_SAS.1		X					
FPT_PHP.3			X				
FDP_ITT.1				X			
FPT_ITT.1				X			
FDP_IFC.1				X			
FDP_SDI.2			X				
FCS_RND.1					X		
FCS_COP.1 – iteration-1 (DES)							X

FCS_COP.1 – iteration-2 (3DES)							X
FCS_COP.1– iteration-3 (AES)							X
FCS_COP.1 – iteration-4 (RSA)							X
FCS_CKM.1/RSA							X
FPT_TST.2						X	