AE55C1 (HD65255C1) Version 02

with ACL Version 1.43

Smartcard Security Target

- Public Version -

Renesas Technology Corp.

Revision 5.0

7 April, 2006

RENESAS

Revision

Revision	Date	Section	Description
Rev. 1.0	21 July, 2005	—	Release
Rev. 2.0	12 December, 2005	0.4	Reference information is updated.
		1.1	Document versions are updated: [ST], [UGM], [ACLSM], [OPT].
		2.1.1	A HW RNG description is modified.
		5.1.1.6	A PRNG description is modified.
		6.1	A small modification in the definition of SF.EEPAccess
Rev. 3.0	19 January, 2006	0.4	Reference information is updated.
		1.2	Document versions are updated: [UGM] and [ACLSM]
Rev. 4.0	30 Jan. 06	0.4	Reference information is updated.
		1.2	Document version and date are updated: [UGM]
Rev. 5.0	7 Apr., 2006	0.4	Added [FIPS], [ISO], and [SHA256] to the reference document list. Updated [OPT] information.
		1.1	Updated the version of this document and the version of referenced full-ST.
		1.2	Added SHA-256 software and SHA-256 manual in Table 1-1, Figure 1-1, and descriptions below the figure.
		2.1.1	Modified the RNG description.
		2.1.2 (inc. Fig. 2-2), 3.4.2, 4.1.2, 5.1, 8.2	Added SHA-256
		5.1.1.6	Small modifications in the texts of FCS_RND.1.1 [TRNG], [PRNG].
		5.1.2.1, 8.2 (T8-3)	Replaced the standards referenced for FCS_COP.1 [SHA-1] and FCS_COP.1 [RIPEMD-160]
		5.1.2.1, 5.2.1, 6.2, 8.2, 8.2.1, 8.3	Added FCS_COP.1 [SHA-256] (in section 6.2: Table 6-1, in section 8.2: Table 8-2 and Table 8-3, in section 8.2.1: Table 8-5, in section 8.3: Table 8-6)
		6.1	SF.Hash: Added SHA-256 and changed the standards to be referenced.
		8.2	Deleted security requirements for the environment corresponding with the objectives of the TOE (non-environmental).

0   Preface

0.1   Objectives of Document

This document defines the Security Target for evaluation of the Renesas AE55C1 with ACL. It is intended to comply with the BSI-PP-0002 Protection Profile (see [BSI-PP-0002]). Although no evaluated conformance claim is made, it is also intended to conform as far as possible with the requirements of the IC package of the SCSUG Protection Profile, and to provide support for operating system and application software that aims to meet SCSUG requirements.

0.2   Scope of Document

This document defines the Security Target as required by [CC/1]. This is a Public version of the document used in the evaluation process.

0.3   Intended Readership

- Developers concerned with certification of the TOE.

- Developers of Smartcard Embedded Software to run on the TOE.

- Evaluators and certifiers of the TOE.

0.4   Related Documents

A reference of the form [REF, n] refers to section n of REF.

Literature

[BSI-PP-0002]   Smartcard IC Platform Protection Profile, v1.0, Eurosmart, July 2001

[AIS31]   Functionality classes and evaluation methodology for physical random number generators, AIS 31, v3.1 (part of Bundesamt für Sicherheit in der Informationstechnik Application Notes and Interpretation of the Scheme), Certification Body of the BSI

[AIS20]   Functionality classes and evaluation methodology for deterministic random number generators, AIS 20, v1 (part of Bundesamt für Sicherheit in der Informationstechnik Application Notes and Interpretation of the Scheme), Certification Body of the BSI

[FIPS]   Secure Hash Standard, Federal Information Processing Standards Publication 180-2, 2002

[ISO]   ISO/IEC FDIS 10118-3, 2003: Information Technology – Security techniques – Hash-functions – Part 3

[CC/1]   ISO/IEC 15408-1:1999 (E) Information Technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model, 1999-12-01

[CC/2]   ISO/IEC 15408-2:1999 (E) Information Technology – Security techniques – Evaluation criteria for IT security – Part 2: Security functional requirements, 1999-12-01

[CC/3]   ISO/IEC 15408-3:1999 (E) Information Technology – Security techniques – Evaluation criteria for IT security – Part 3: Security assurance requirements, 1999-12-01

Note: the combination of all 3 parts of ISO 15408 is also referred to in this document as "Common Criteria" or "CC".

[PA]   Smartcard Integrated Circuit Platform Augmentations, v1.0, Atmel, Hitachi Europe, Infineon Technologies & Philips Semiconductors, March 2002

Documents and User Guidance

[HM]	Renesas 32-bit Smart Card Microcomputer AE-5 Series AE55C1 (HD65255C1) Hardware Manual, Rev. 1.00, Renesas Technology Corp., 15 March 2005
[OPT]	Option List for Smart Card Microcomputer (for HD65255C1 [AE55C1]), v.1.3, Renesas Technology Corp., 6 February, 2006
[PM]	Hitachi Single-Chip Microcomputer H8SX Programming Manual, Revision 1.0, Hitachi Ltd., February 2003
[ACLSM]	AE-5 Series Cryptographic Library User's Manual, Rev. 1.70, Renesas Technology Ltd – Engineering Division, 27 January, 2006
[UGM]	AE-5 Series User Guidance Manual, Rev. 4.40, Renesas Technology Corp., 27 January 2006
[SHA256]	AE-5 Series Cryptographic Library, Renasas 32-bit Smart Card Microcomputer AE-5 Series, Information for software developers using the SHA-256 with the Advanced Cryptographic Library, Revision 1.01, Renesas Technology Europe Ltd., 15 March 2006

0.5   Document Structure

This document follows the structure set out in [CC/1].

Note that sub-sections are structured to show parts that originate from [BSI-PP-0002], [PA] and other sources.

0.6   Outstanding Issues

None.

0.7   Glossary

Term	Meaning
ACL	Advanced Cryptographic Library
CBC	Cipher Block Chaining A mode of DES encryption.
CC	Common Criteria (ISO 15408)
COT	Chip-on-Tape - an IC packaged in a form suitable for embedding into a plastic card to form a smart card.
CPU	Central Processing Unit
CRAM	RAM dedicated for use by MMC coprocessor
CRC	Cyclic Redundancy Check
CRT	Chinese Remainder Theorem
DES	Data Encryption Standard
DMAC	Direct Memory Access Controller
DFA	Differential Fault Analysis
DPA	Differential Power Analysis
EAL	Evaluation Assurance Level
ECB	Electronic Code Book A mode of DES encryption.
EEPROM	Electronically Erasable Programmable Read-Only Memory

Term	Meaning
Embedded Software	Software held in the chip, having been developed by users of the TOE. Such software generally includes an operating system and may include all or part of applications. There are two types of Embedded Software: • Hard-coded – held in ROM • Soft-coded – held in EEPROM or RAM. The chip does not depend on such software, and Embedded Software is not part of the TOE. However, hard-coded Embedded Software will be included in the ROM of any issued TOE at manufacturing time. Note that Embedded Software includes all software on a TOE other than the IC Dedicated Software. Embedded Software is also referred to as 'Smartcard Embedded Software' (especially in [BSI-PP-0002]).
EWE	An interrupt generated by the TOE whenever an attempt is made to write to EEPROM.
FMU	Firewall Management Unit – a feature of the TOE that limits the memory areas available.
HW RNG	Hardware random number generator (physical random number generator)
IC	Integrated Circuit
IC Dedicated Software	Software developed by Renesas and embedded in the IC. (Adopted from [BSP-PP-0002])
IC Dedicated Test Software	Software developed by Renesas for testing the TOE during manufacture. This software is part of the TOE, but is not available for general use by operating systems, applications or end-users in phase 7 of the lifecycle (see section 2.3).
ISC	Integrated Security Concept
IT	Information Technology
LSI	Large Scale Integration
Manufacturing Identification Data	Some basic data injected into EEPROM, enabling traceability of an IC to the lot and line in which it was manufactured, the Smartcard Embedded Software present, and the versions of masks and specifications applicable.
MCU	Micro Computer Unit
MMC	Modular Multiplication Coprocessor
Option List	A form supplied by Renesas and filled in by a TOE customer, specifying various options for the manufacture of TOE ICs for that customer. The aspect of particular interest to this security target is : • Selection of whether pre-personalisation data injection is required The option list also describes the content and structure of the manufacturing identification data that Renesas will inject (see section 2.4.2).
PKI	Public Key Infrastructure
PP	Protection Profile
PRNG	Pseudo Random Number Generator
RAM	Random Access Memory
Renesas	Refer to Renesas Technology Corp. (http://www.renesas.com/)
Reset state	A state in which the chip does not execute instructions or engage in input/output. The chip can exit the reset state by receiving an external reset. See also section 6.1.
RNG	Random number generator
ROM	Read-Only Memory
RSA	Rivest, Shamir, Adelmann – a public key encryption algorithm, named after its inventors.
SFP	Security Function Policy
SFR	Security Functional Requirement
Smartcard	(as used in [BSI-PP-0002]) Composition of the TOE, the Smartcard Embedded Software, User Data and the package (the smartcard carrier).
SOF	Strength of Function
ST	Security Target
TOE	Target of Evaluation. TOE consists of the chip (IC) and other materials and data. However, the term is sometimes used to indicate just the chip.
TOE Delivery	The point at which the TOE is delivered, as shown in section 2.3. This may be either in the form of wafers (at the end of phase 3) or as packaged modules (at the end of phase 4).

Term	Meaning
TOE Manufacturer	(As defined in section 8.7 of [BSI-PP-0002]) The IC developer and manufacturer. If the TOE is delivered after phase 4 (i.e. as packaged modules, rather than wafers) then this is also the packager. For the AE55C1 with ACL, the TOE Manufacturer refers to Renesas.
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSF Data	Data created by and for the TOE and that might affect the operation of the TOE.
UART	Universal Asynchronous Receiver Transmitter – in accordance with ISO/IEC7816-3.
User Data	All data managed by the Smartcard Embedded Software in the application context. User data comprises all data in the final smartcard IC except for the TSF data.
User Mode	The mode of operation after TOE Delivery. The TOE is set to this mode just before delivery, which renders the IC Dedicated Test Software permanently unavailable.
WAP	Wireless Application Protocol
WDT	Watchdog Timer – a feature of the chip that enables embedded software to be regularly executed during the operation of the IC. This allows checks to be made on the execution environment to help detect potential attacks or insecure conditions.

Table of Contents

1.   ST Introduction

The ST aims to provide potential users of the TOE with

- A definition of the main properties of the IC that are evaluated and certified independent of any software

- Confidence in IC properties that can be used to build an integrated TOE (i.e. IC + operating system + other application software).

1.1   ST Identification

Title: AE55C1 (HD65255C1) Version 02 with ACL Version 1.43 Smartcard Security Target – Public Version

Revision: 5.0

Provided by: Renesas Technology Corp.

This Security Target applies to the Renesas AE55C1 integrated circuit (version 02) with ACL (version 1.43)
(as defined in detail in the configuration list for the evaluation).

This public version of the AE55C1 Security Target (also known as "ST-lite") is abridged from the evaluated version of the full Security Target (revision 15.0) with the approval of the Certification Body.

1.2   TOE Identification

The TOE configuration is summarised in the table below:

Table 1-1: TOE Configuration

Item Type	Item	Version	Form of delivery
Hardware	AE55C1 (HD65255C1) smartcard integrated circuit	02	Wafer or packaged module (see section 2.3)
Software	IC Dedicated Test Software Test ROM software	1.4	Included in AE55C1 test ROM
Software	ACL (Advanced Cryptographic Library)	1.43	Software module (This is implemented in the Embedded Software by the user)
Software	SHA-256 software	1.10	Software module (This is implemented in the Embedded Software by the user)
Document	Hardware Manual	1.00	Hardcopy
Document	User Guidance	4.40	Hardcopy
Document	Cryptographic Library Manual	1.70	Hardcopy
Document	Software Manual for SHA-256	1.01	Hardcopy
Document	Option List	1.3	Electronic data/Hardcopy

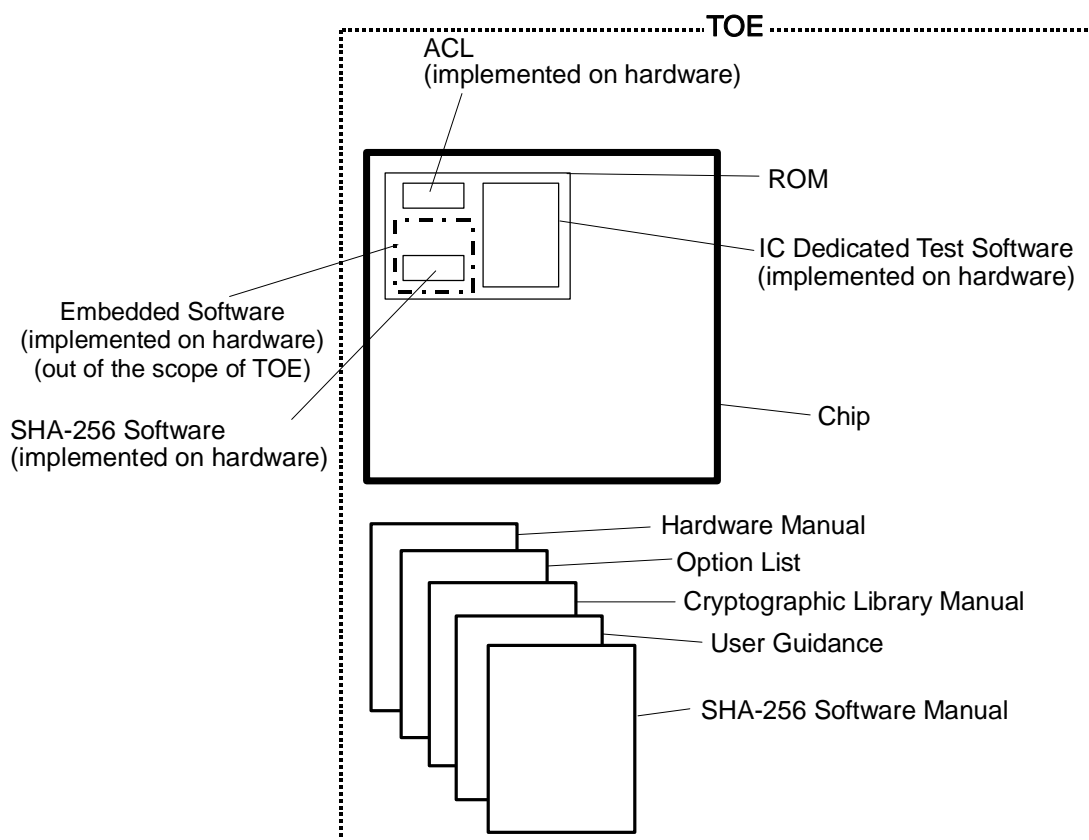Further description of the TOE is provided in section 2.1.

Figure 1-1: Configuration of the TOE

Figure 1-1 shows the configuration of the TOE. The TOE consists of hardware, software, and documents. The software (as part of the TOE) is provided as being implemented on the hardware. The ACL (Advanced Cryptographic Library) is provided to the user as an object code in an electronic form, and it is implemented in the TOE as the embedded software by the user. The SHA-256 software is also provided to the user as an object code in an electronic form, and it is implemented in the TOE as the embedded software by the user.

1.3 ST Overview

This document is the ST for the Renesas AE55C1 IC product with ACL, intended for use as a smart card IC. The TOE complies with the Eurosmart Protection Profile developed by the Secure Semiconductor Vendor Group [BSI-PP-0002]. However, the TOE also provides a number of additional security features that have been based on a long history of assisting software developers to implement secure Smartcard Embedded Software. These TOE-specific security features have been added to the ST.

The TOE is ideally suited for high security applications. Security has been built in from the start, forming an integral part of the whole Smartcard design concept. The whole development process (including secure chip design environment, secure production facilities and secure handling during shipment to the customer) is constantly reviewed in order to maximise the overall security package. The TOE can be delivered either in the form of wafers, or as packaged modules ready for embedding into a plastic card.

Many security features such as integrated sensors, distributed layout, random number generation, DES engine and power analysis attack protection are all included providing a strong on-chip hardware security structure.

Uniquely, Renesas smartcard devices are fabricated using a MONOS (Metal Oxide Nitride Oxide Silicon) EEPROM structure. MONOS advantages compared to standard EEPROM structures are high resistance to radiation disturbance, high reliability and endurance.

The TOE fulfils the requirements of Smart Card applications requiring large memory, high security and high speed secure authentication, data encryption or electronic signature. Examples include: PKI, WAP, e-m-commerce, digital signature, USIM/UMTS, and credit card.

Where public key is a core requirement, a high speed MMC able to process arithmetical data in a time frame that ensures a fast and free flowing application environment is required. The MMC ensures the high performance required by today's high security applications.

Applications such as WAP and e-m-commerce are ever expanding in scope and consequently the need for greater memory storage for both data and program code is ever increasing. The TOE provides a significant increase in ROM for program storage over previous devices whilst ensuring a balance of EEPROM for data storage.

The move from single to multi-application on a single component is also rising due in part to new systems such as WAP and e-m-commerce. This requires not only additional memory for application data storage but also features such as FMU in order to provide data integrity between applications.

1.4   CC Conformance Claim

This ST is compliant with [CC/1], [CC/2] and [CC/3].

Because the ST conforms to [BSI-PP-0002], it includes extended functionality classes defined in section 8 of [BSI-PP-0002]. The ST is therefore [BSI-PP-0002] conformant, [CC/2] extended and [CC/3] conformant. In addition, this ST includes some additional assumptions, threats, objectives and SFRs defined in [PA]. Therefore, this ST is also [PA] conformant.

The Assurance level is EAL4 augmented (for augmentations see section 5.1.4).

The strength of function is SOF-high.

2. TOE Description

2.1   AE55C1 with ACL Product Description

The TOE consists of the hardware shown in Figure 2-1, along with IC Dedicated Test Software, some embedded software, and reference and guidance documents. IC Dedicated Test Software is used in IC production only, and is not available to users.

As well as the functional interfaces, the IC surface is also considered as a TOE interface for some potential physical attacks, as described in section 3.3 of [BSI-PP-0002].

A block diagram of the chip is shown in Figure 2-1 below:
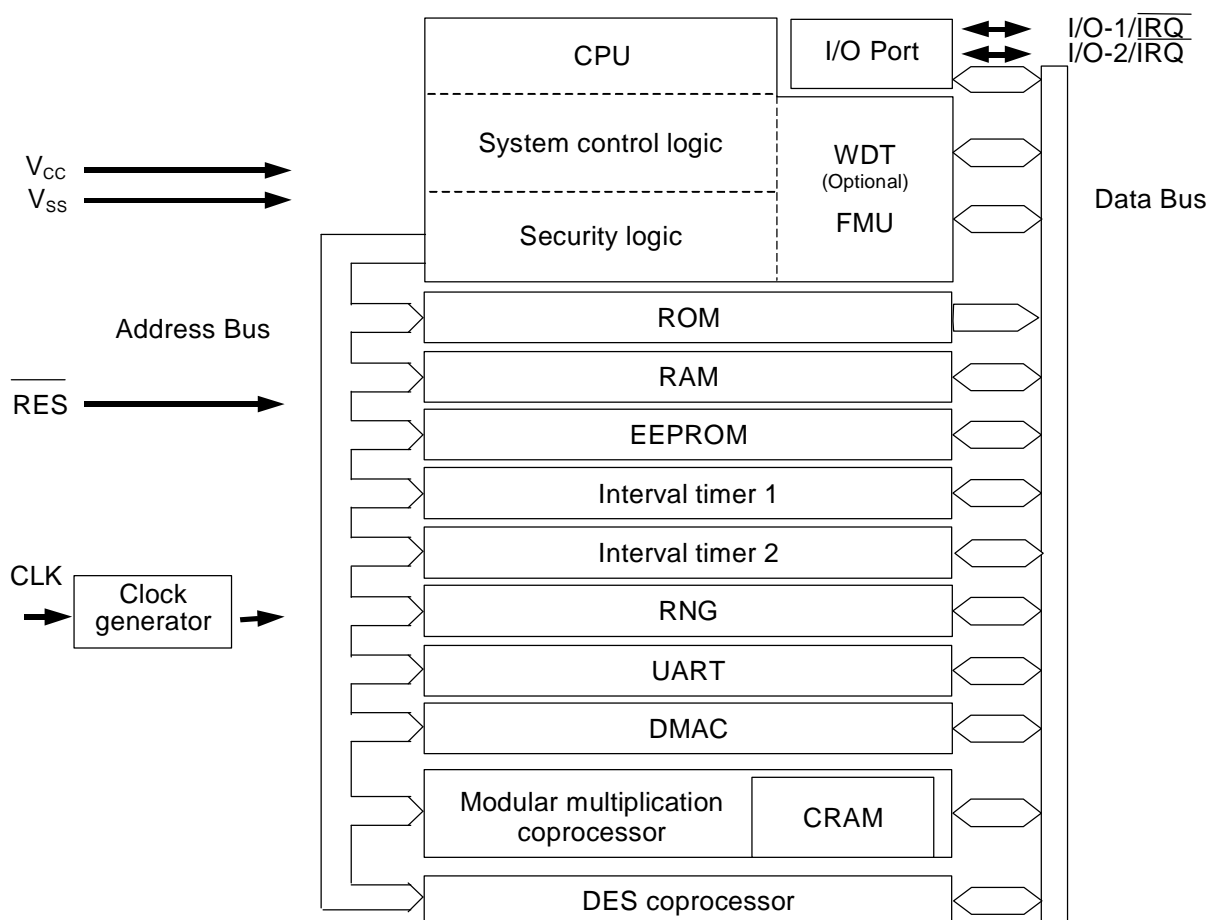
Figure 2-1: Internal Block Diagram of the TOE

2.1.1   Hardware

The TOE is an integrated circuit for smartcard based around the high-speed AE-5 Series CPU core (derived from Renesas' widely used H8SX general purpose core). As can be seen from the block diagram in Figure 2-1, the MCU comprises the following major blocks in addition to the CPU: ROM, RAM, EEPROM, random number generator (RNG), MMC, DES coprocessor, DMAC, UART, two interval timers, WDT (optional), FMU, and two I/O lines.

CPU: the CPU can operate with either a 3V or 5V supply and a maximum internal clock frequency of 20MHz (at 3V) and is upwards compatible with the H8/300 core (as used in Renesas' AE-4 series of smartcard ICs):

- The instruction set is implemented with 16-bit variable instruction lengths (2 to 14 bytes) and permits register to register arithmetic and logic operations.

- A linear address space of up to 16 Mbytes is possible.

- Signed or unsigned multiply instruction (8 x 8, 16 x 16, or 32 x 32-bits)

- Signed or unsigned divide instruction (16 ÷ 8, 16 ÷ 16, 32 ÷ 16, 32 ÷ 32-bits)

- Special EEPROM write instruction and high-speed block transfer instruction

Memory: the TOE has a memory mapped architecture and allows the EEPROM to be used for both data and program storage. There is a special block of RAM dedicated for coprocessor use, that can be used as normal RAM when not required by the coprocessor.

- ROM: 240 kbytes (for Embedded Software)

    16 kbytes (for IC Dedicated Test Software)

- RAM: 6 kbytes + 2 kbytes of coprocessor RAM (CRAM)

- EEPROM: 32 kbytes + 8 kbytes

    o on-chip charge pump and independent oscillator

    o special write instruction, and interrupt generation on writing

    o page write/erase

    o individual page protection

RNG: this can generate 64-bit random numbers. Using the RNG enables a unique value to be generated inside the chip, which improves the system security.

MMC (Modular Multiplication Coprocessor): this unit consists of the control unit and the 2k bytes CRAM. The control unit is intended to form the basis for efficient implementations of asymmetric cryptography by providing high speed modular multiplication in hardware with a programmable data length from 160 – 2112 bits.

DES coprocessor: this hardware engine can be used to provide either DES or triple-DES functions to the users' software with very little software overhead. Countermeasures against information leakage have been integrated into the coprocessor unit to make it highly resistant to such attacks with minimal software overheads or execution time penalties. These countermeasures are always active and make no additional requirements on Smartcard Embedded Software.

Interval timer: the TOE has interval timer 1 and interval timer 2. These timers are identical and issue an interrupt at intervals which user determined.

WDT: the watchdog timer is a powerful tool to help the user software detect and respond to unauthorised program execution. Use of the WDT is optional and should be selected using [OPT].

FMU: this memory management function monitors memory access permissions for the user software and enables the limiting of program execution from RAM and EEPROM.

I/O: I/O comprises I/O-1/IRQ and I/O-2/IRQ (see Figure 2-1). As well as the ISO 7816 standard I/O pin, a further I/O pin is provided for additional use. These pins, together with the power and clock pins, form the electrical interface of the TOE.

UART: half-duplex asynchronous mode that conforms to the ISO/IEC standard 7816-3. Full-duplex asynchronous mode that conforms to the ISO/IEC CD standard 10536-4:

DMAC: this transfers data at high-speeds between the RAM and the UART, relieving the CPU of this function. The DMAC can also clear a specified area of RAM to 0 without using the CPU.

System control logic: System control logic generates a signal to control the interface between the CPU subsystem and each other subsystem.

Security logic: the IC incorporates specialised security logic to help to ensure the correct operation of the TOE.

Some security features, such as FMU, allow the Smartcard Embedded Software to determine the events to be monitored and the actions to be taken; many of them however are independent of software and are designed to operate automatically when required, to return the TOE to a secure state. The IC also provides protection features to resist leakage attacks.

Physical security of the IC is enhanced by the presence of shielding over critical areas, and by the use of design techniques that obscure the function and operation of the physical layout.

Full details of the operation of the chip and guidance for its use are given in [HM], [OPT], and [UGM]. Guidelines provided specifically for use of ACL is [ACLSM].

2.1.2   Software

The TOE includes the following software:

IC Dedicated Test Software:

The IC Dedicated Test Software is integrated into the TOE hardware. It is used for mode transition and testing during IC production, and is not available to users.

ACL (Advanced Cryptographic Library):

The ACL provides the following functions:

- RSA: Secure implementation of the standard RSA and RSA CRT algorithm[1].

---

[1] The RSA function supports the encryption key lengths from 512-bit to 2048-bit, but only the lengths from 1024-bit to 2048-bit are claimed as a part of Renesas' TSFs.

- Secure Modular Inversion: securely calculate the modular inversion of a big number.

- Key Generation: Secure implementation of key generation for RSA and RSA CRT.

- Online Tests: secure implementation of online tests of the underlying platform's hardware random number generator.

- Random Number Generator: Secure implementation of a software random number generator providing high quality pseudo random numbers.

- CRC functions: The cryptographic library contains two functions that generate a CRC value of a byte string. The CRC16 is implemented according to the CRC-CCITT algorithm. The CRC32 is built according to a widely used CRC polynomial function[2].

- Scrambled copy / compare / XOR: to allow the user software to copy, compare or XOR data blocks containing secret information.

- Secure multiplication: to allow the multiplication of data blocks containing secret information, the ACL implements a secure multiplication function on the hardware MMC.

- DES/3DES: secure implementation of the DES and 3DES algorithms using the hardware DES engine.[3]

- Hash functions: secure implementation of the hash functions SHA-1, SHA-256 and RipeMD-160,

The ACL does not include any key management functionality, other than RSA key generation, since this depends on the application context.

The TOE is able to select and combine appropriate co-processor security functions and to provide an interface to perform basic cryptographic functions (as listed above).

All other Smartcard Embedded Software (e.g. an operating system) is outside the scope of the TOE. The Smartcard Embedded Software is supplied to Renesas by the customer in a secure manner, and is then protected by Renesas' secure production environment.

The relationship between the ACL, the TOE Hardware, and the customer application (out of the scope of the TOE) is illustrated in Figure 2-2:

---

[2] The CRC is an effective security function, but it is not claimed as a part of Renesas' TSFs.

[3] Though the TOE has both the single DES function and the triple DES function, it is only the triple DES function that is claimed as a part of Renesas' TSFs.
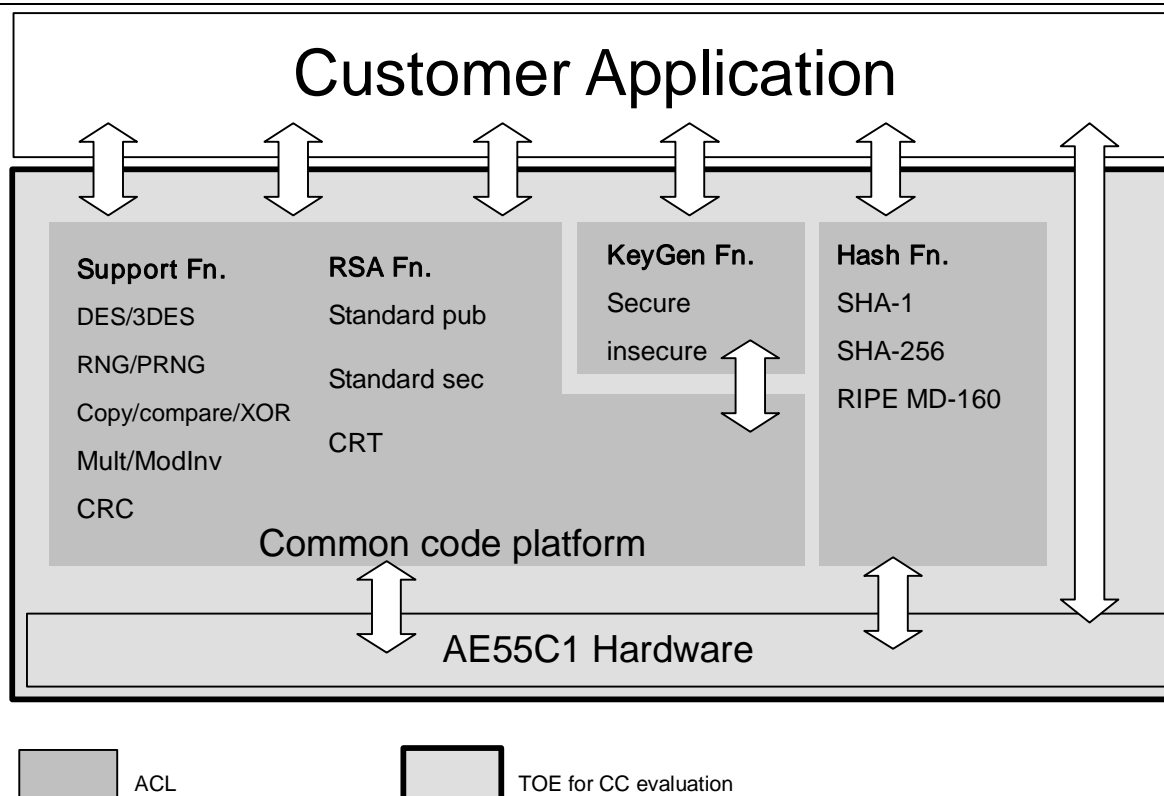
Figure 2-2: ACL (TOE Software), TOE Hardware and Customer Application

2.1.3 Documents

The TOE Hardware Manual [HM] and TOE ACL User Manual [ACLSM] are supplied as the basic reference for users who are developing Smartcard Embedded Software. Guidance for the secure use of the AE-5 Series in applications is given in the User Guidance Manual [UGM], and [ACLSM]. Options and contents of the identification data stored in the EEPROM are described in [OPT].

2.2 TOE Intended Usage

The TOE is intended for use in a range of high security applications, including high speed secure authentication, data encryption or electronic signature. Examples include: PKI, WAP, m-commerce, digital signature, USIM/UMTS, and banking card.

2.3 TOE Lifecycle

The design and manufacturing lifecycle for the TOE is shown in Figure 2-3 (and in section 8.1.1 of [BSI-PP-0002]). The TOE can be delivered either at the end of phase 3, or at the end of phase 4, as shown in the figure.
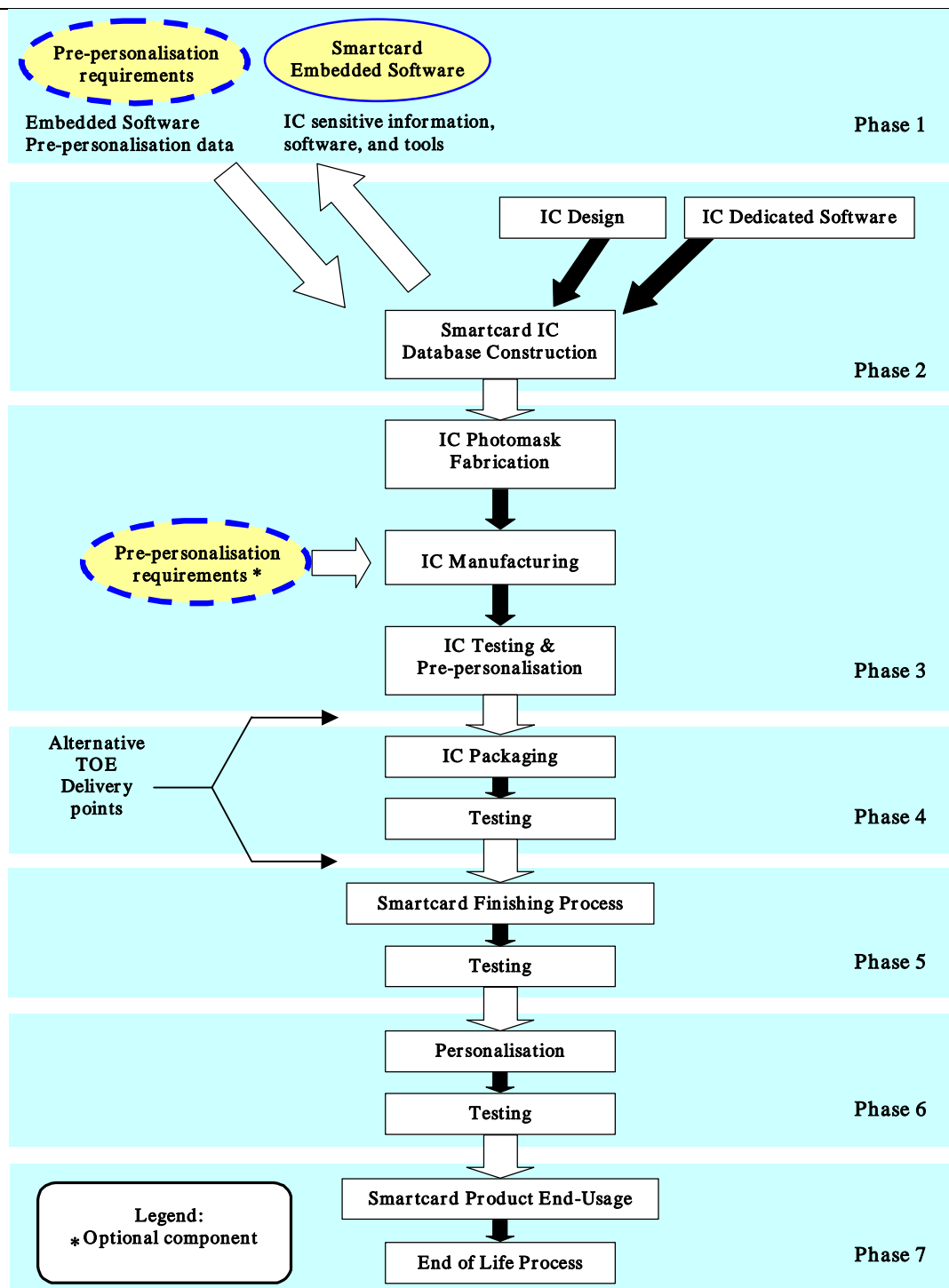
Figure 2-3: Design and Manufacturing Lifecycle

The stages shown are as listed below. This Security Target addresses phases 2 and 3 in Figure 2-3. When the TOE is delivered by Renesas as modules rather than in wafer form, phase 4 is also covered under this Security Target.

- Phase 1: Smartcard Embedded Software development – this phase is outside the scope of the TOE, but the results of the software development are inputs to the manufacture of a customer-specific instance of the TOE. Renesas deliver information about the TOE (such as [HM]) to the embedded software developer to enable the Smartcard Embedded Software to be written. Development tools (such as emulators) and IC samples are also available to developers.

Information and tools are released only under Non-Disclosure Agreement to ensure their distribution is controlled and limited (this ensures, for example, secure disposal of scrap).

The embedded software for incorporation in the ROM mask is sent via a secure delivery method from the software developer to Renesas, and its secure handling is then ensured by Renesas. Similarly, pre-personalisation data is sent by a secure route to Renesas for injection during the manufacturing stage. Injection of data is described further in section 2.4.2. Secure receipt and handling of this data by Renesas is included within the scope of this security target.

- Phase 2: IC design, IC Dedicated Test Software and ACL software development – this includes system design through logic, circuit and layout design. The Dedicated Test Software and ACL software are also developed in this phase, to enable testing of the IC at various phases of its manufacture. The Smartcard Embedded Software is received from phase 1. Meanwhile, the object code of the ACL software is sent to phase 1.

  The security of the development environments for the IC and its dedicated software, along with the secure handling of masks are primary concerns of this security target.

- Phase 3: Mask fabrication and IC wafer manufacturing – The masks used to manufacture the various IC layers are created from the layout design during phase 2; ROM masks for the Smartcard Embedded Software in a particular instance of the TOE are fabricated after receipt of the software in phase 2. Also, in this phase, wafers containing TOE dies with Smartcard Embedded Software are produced. At the end of this process each die is tested and has its pre-personalisation data injected into EEPROM (see section 2.4.2).

  The secure fabrication and handling of masks, and the security of manufacture, including handling of masks, test software, and pre-personalisation data are primary concerns of this security target.

  The TOE may be delivered at this stage as wafers (in which case each die will be set to User Mode). Alternatively, the TOE may be packaged by Renesas and delivered in COT form at the end of phase 4. If the TOE is to be delivered in COT form then at the end of phase 3 it is set to a constrained Test Mode, instead of User Mode, to allow testing of the IC after module manufacturing in phase 4.

  This security target covers TOE Delivery at either point, but where wafers are delivered at the end of phase 3, secure handling in phase 4 is a customer responsibility. Note that in all cases the TOE will be set to User Mode before TOE Delivery.

- Phase 4: IC Packaging – each wafer is cut into individual dies and its protective packaging applied, along with its contacts. The resulting module is then tested in Test Mode to ensure correct operation, after which it is set to User Mode. This leaves the IC ready for insertion into a plastic card.

- Phases 5-7: Smartcard finishing, personalisation and end-usage – these stages, and their security features, are determined by the customer, and are outside the scope of this security target.

2.3.1   Test and User Modes

During manufacturing and production (phases 2-4), the chip makes some mode transitions which affect the interface presented. These modes are as follows:

- Test mode makes the IC Dedicated Test Software available, which is used to ensure that only correctly working ICs are delivered.

- User mode makes the IC Dedicated Test Software unavailable and the TOE now provides only the functionality described in [HM]. The software interface presented will be determined by the Smartcard Embedded Software.

As explained under the individual lifecycle phases above, the chip is always in test mode in phases 2 and 3. If TOE Delivery takes place at the end of phase 4, then the chip will be in the constrained test mode during phase 4, making a limited set of test functions available to test the correct operation of modules after packaging. However, the TOE is always in User Mode at the point of TOE Delivery, and the transition to User Mode is irreversible.

2.3.2   TOE Delivery

As noted above, the TOE may be delivered at the end of either phase 3 or phase 4, as requested by the customer. In either case, the chip will be delivered in User Mode, and Renesas will apply secure delivery procedures for the transport of the TOE from Renesas premises.

2.4   TOE Environments

2.4.1   Development Environment

Renesas' development environment for the TOE has implemented security measures specifically to ensure the security of the TOE and of Smartcard Embedded Software and injection data used in manufacturing ICs for customers. As indicated in section 2.3, there are three areas of the development environment:

- Design sites

- Mask manufacture site

- Manufacturing sites

These provide the following main security properties:

- Design sites

  - Confidentiality and integrity of logical and physical design

  - Testing of TOE security functionality

  - Confidentiality and integrity of IC Dedicated Test Software

  - Confidentiality and integrity of ACL (object code and source code)

- Confidentiality and integrity of customer ROM code and injection data (i.e. the Smartcard Embedded Software for an instance of the TOE)

- Mask manufacture site

  - Confidentiality and integrity of customer ROM code and mask

  - Confidentiality and integrity of design and base masks

- Manufacturing sites

  - Confidentiality and integrity of base masks

  - Confidentiality and integrity of customer ROM mask

  - Confidentiality and integrity of test software

  - Confidentiality and integrity of injection data

  - Production of authentic TOE ICs, correctly implementing the design and including the customer Smartcard Embedded Software in ROM.

Security issues for each of these areas are addressed by processes and procedures put in place by Renesas and within the scope of evaluation. The security measures include IT security to protect information stored on Renesas computer systems, as well as physical security measures for secure storage to ensure that design and manufacturing information and objects are only accessible to authorised staff with a need to know the information. Renesas' integrated security concept (ISC) covers the entire development process, from specification, through design and implementation to manufacturing and shipping. ISC is implemented through the use of standards and procedures that form part of the quality system at the heart of Renesas' business. The rigorous adoption and adherence to procedures, including those relating to security, is an integral part of the quality system at the heart of Renesas' business.

The security of the IC Dedicated Test Software and ACL software at design and manufacturing sites is ensured by the same level of security measures as for the hardware design. This ensures that only authorised persons have access to the software and its related information.

Smartcard Embedded Software is received from customers via a secure delivery procedure. Once received by Renesas, this software is also handled with the same level of security as for design information. As a further measure, the group handling customer software is separate from the IC design team.

2.4.2 Injection of Manufacturing Identification and Secret Data

In general, although there will be a substantial amount of operating system and application software held in ROM, an IC will also require software to be added after it leaves the manufacturing environment. Operating system software may require additional parts or patches to be loaded, and increasingly applications are expected to be loaded and deleted after a smart card has been issued to users. In order to enable such addition (and deletion) of software to be done securely, there is a generic requirement for identification data and secret data, determined by the IC purchaser, to be injected during manufacture.

The TOE supports this requirement by injecting identification data during the manufacturing process; this data uniquely identifies each IC. In addition, customers may choose to inject further data. The details of the data content and its location in memory are shown in [OPT].

3.   TOE Security Environment

3.1   Assets

This section defines the primary and secondary assets to be protected by the TOE. Section 3.1 of [BSI-PP-0002] gives the assets relating to the threats, and these are summarised below.

The primary assets to be protected are classified as:

- User Data – this includes injection/pre-personalisation data and data generated and managed by the Smartcard Embedded Software (subject to adequate protection by the software, see A.Key-Function, A.Plat-Appl and A.Resp-Appl in section 3.2)

- Smartcard Embedded Software, comprising

  - Hard-coded Embedded Software – this is fixed and generally consists of parts or all of the operating system, and parts or all of a number of applications

  - Soft-coded Embedded Software – this may include parts of the operating system or applications.

Both of these types of asset need to have their confidentiality and integrity protected.

A further primary asset is:

- Correct operation of the TOE (including its random number generator).

In particular this means that the Smartcard Embedded Software will be correctly executed, which includes the correct operation of the TOE's functions.

Because random numbers are likely to be used by embedded software for generating cryptographic keys, another primary asset is:

- the random numbers generated by the TOE[4]

Secondary assets are critical information about the TOE, including logical design data, physical design data, IC Dedicated Test Software and TSF data.

In addition, the following will also contain information about the TOE.

- Initialisation Data and Pre-personalisation Data, specific development aids, test and characterisation related data, material for software development support, and photomasks.

Such information and the ability to perform manipulations assist in threatening the primary assets.

---

[4] The confidentiality of random numbers is generally protected by embedded software (which is responsible for requesting random numbers). However, it is important that random numbers should not be subject to leakage (cf. T.Leak-Inherent), because of their potential role in cryptographic key generation.

Assets relating specifically to the Organisational Security Policy P.Process-TOE and Assumption A.Process-Card are given in section 3.1 of [BSI-PP-0002].

3.2   Assumptions

3.2.1   Assumptions from [BSI-PP-0002]

The following assumptions are made:

A.Process-Card   Protection during Packaging, Finishing and Personalisation

It is assumed that security procedures are used by the recipient after delivery of the TOE by Renesas up to delivery to the end-user 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).

The exact requirements of this assumption will depend on the Smartcard Embedded Software. This means that the phases after TOE Delivery (refer to section 2.3) are assumed to be protected appropriately. For a preliminary list of assets to be protected, see section 3.1 of [BSI-PP-0002].

A.Plat-Appl   Usage of Hardware Platform

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

(i)   The TOE hardware manual [HM], user guidance manual [UGM], ACL User Manual [ACLSM], and the hardware application notes

(ii)   Findings of the TOE evaluation reports relevant for the Smartcard Embedded Software.

A.Resp-Appl   Treatment of User Data

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

This assumption requires that the Smartcard Embedded Software define and positively manage its security relevant User Data, in the manner required by the application context. Without this, the protection provided by the TOE itself may be of no use if the Smartcard Embedded Software itself allows data to be compromised.

Examples of embedded software security concerns are given in section 8.2 of [BSI-PP-0002].

3.2.2   Assumptions from [PA]

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 (i) the cryptographic routines which are part of the TOE and (ii) the processing of User Data including cryptographic keys.

Note here that the functions considered in this assumption are part of the Smartcard Embedded Software; T.Leak-Inherent and T.Leak-Forced address the cryptographic functions that are part of the hardware.

For example: consider the RSA encryption system implemented in the ACL. This uses the modular arithmetic functions of the MMC, and the RSA functions in the ACL. In this case the leakage characteristic of the implementation will depend partly on the hardware characteristics of the TOE and its MMC, partly on the way the Software part of the TOE (ACL) uses this hardware to implement RSA encryption and partly on the ways in which the embedded software calls this RSA encryption function and how it provides the inputs and uses the results. The properties of the TOE are assessed under this Security Target, but the security properties of the embedded software are clearly outside the scope of the TOE evaluation.

To assist embedded software developers to implement leak-resistant code, guidance on secure software implementation is given in [UGM] and [ACLSM].

3.2.3   Other Assumptions

A variety of keys and other security-critical data may be injected for use by Smartcard Embedded Software. These may include shared private keys, public/private key pairs, etc. This information could contribute to a cloning attack, or to breaking the security of an instance of the TOE (e.g. by compromising its keys). The integrity of this data is also vital to ensuring the security of the TOE (e.g. preventing unauthorised changes to mask code, IC design or keys). All data supplied for injection/pre-personalisation is assumed to be supported off-card in a secure manner:

A.InjDatSupp   Injected Data Support

Data for injection/pre-personalisation will be supplied from the various bodies controlling the operations of the system in which the TOE is functioning, based on [OPT]. It is assumed that the generation, distribution, maintenance, and destruction of this data is adequately secure.

3.3   Threats

3.3.1   Threats Defined in [BSI-PP-0002]

This section adopts the threats to ICs defined in section 3.3 of [BSI-PP-0002].

The TOE has the following high-level security concerns, as in section 3.3 of [BSI-PP-0002]:

SC1   manipulation of User Data and of the Smartcard Embedded Software (while being executed/processed and while being stored in the TOE's memories).

SC2   disclosure of User Data and of the Smartcard Embedded Software (while being processed and while being stored in the TOE's memories).

SC3   deficiency of random numbers.

These high-level security concerns are refined below by defining specific threats. Note that manipulation of the TOE is only a means to threaten User Data or the Smartcard Embedded Software and does not in itself represent a successful attack.

These security concerns are derived from considering the end-usage phase (phase 7) since

- Phase 1 and phases from TOE Delivery up to the end of phase 6 are covered by assumptions, and

- The development and production environment starting with phase 2 and ending with TOE Delivery are covered by an organisational security policy

3.3.1.1   Threats Derived from SC1-SC3

See section 3.3 of [BSI-PP-0002], and the example attack scenarios in section 8.3 of [BSI-PP-0002]. For completeness, the threats are summarised below.

T.Leak-Inherent   Inherent Information Leakage

An attacker may exploit information which is leaked from the TOE during usage of the Smartcard in order to disclose confidential data (User Data or TSF Data).

No direct contact with the Smartcard 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.

Differential Power Analysis (DPA) is an example of an attack based on the inherent leakage threat.

T.Phys-Probing   Physical Probing

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

(i)   disclose User Data

(ii)   disclose/reconstruct the Smartcard Embedded Software

(iii)   disclose other critical operational information especially TSF data.

Physical probing requires direct contact with the TOE circuit structures. Before attacks based on this threat can be mounted, 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.

T.Phys-Manipulation   Physical Manipulation

An attacker may physically modify the Smartcard in order to

(i)   modify security features or functions of the TOE

(ii)   modify security functions of the Smartcard Embedded Software

(iii)   modify User Data.

Modification may be achieved through techniques commonly employed in IC failure analysis and IC reverse-engineering. The modification may result in the deactivation of a security function. As with T.Phys-Probing, before attacks based on this threat can be mounted, hardware security mechanisms and layout characteristics need to be identified. Determination of software design including treatment of User Data may be a pre-requisite. Changes to circuitry or data can be permanent or temporary.

In contrast to malfunctions (see T.Malfunction), the attacker needs to gather significant knowledge about the TOE's internal construction in order to launch a manipulation attack.

T.Malfunction   Malfunction due to Environmental Stress

An attacker may cause a malfunction of the TSF or of the Smartcard Embedded Software by applying environmental stress in order to

(i)   deactivate or modify security features or functions of the TOE

(ii)   deactivate or modify security functions of the Smartcard Embedded Software.

This may be achieved by operating the IC outside its normal operating conditions.

Unlike T.Phys-Manipulation, attacks based on environmental stress can be launched without significant knowledge of the IC's internal construction on the part of the attacker. However, to exploit the attack, the attacker needs information about the functional operation.

T.Leak-Forced   Forced Information Leakage

An attacker may exploit information which is leaked from the TOE during usage of the smartcard in order to disclose confidential data (User Data or TSF Data), 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" (T.Malfunction) and/or "Physical Manipulation" (T.Phys-Manipulation) are used to cause

leakage from signals which normally do not contain significant information about secrets.

Differential Fault Analysis (DFA) is an example of an attack based on the forced leakage threat.

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

(i)   disclose or manipulate User Data

(ii)   manipulate (explore, bypass, deactivate or change) security features or functions of the TOE or of the Smartcard Embedded Software

(iii)   enable an attack.

For the TOE, T.Abuse-Func concerns the threat of unauthorised access to the IC Dedicated Test Software, which is rendered inaccessible by placing the IC into User Mode before TOE Delivery (see section 2.3).

T.RND   Deficiency of Random Numbers

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

An attacker may gather information about the random numbers produced.

Malfunctions or premature ageing are also considered as they may assist in giving the attacker information about random numbers.

Attacks on random number generation are significant because the random numbers generated may be used as secrets - e.g. to generate cryptographic keys.

Under the threat T.RND, the attacker is assumed to take advantage of statistical properties of the random numbers generated by the TOE without specific knowledge about the RNG itself.

Also under T.RND, random number is assumed to include both true random number and pseudo random number.

3.3.2   Other Threats

The TOE makes available facilities that are useful for embedded software to use in addressing the threats that it may face. This introduces an additional high-level security concern for the TOE:

SC4   attacks on the Smartcard Embedded Software may be made, and the software may not be able to detect or respond to the attacks.

T.NoSWDetect   Inability of the TOE to detect an attack

In a multi-layered or multi-application software environment, there may be attacks on one part of the Smartcard Embedded Software arising from another part. Smartcard Embedded Software execution may also be attacked via various means, with the intention of corrupting software execution in a specific or random way. If the TOE cannot detect such attacks, then it cannot apply any countermeasures to protect itself.

The TOE is concerned in particular to detect the following:

(i)   Attempts to access memory outside an area defined for the software being executed

(ii)   Attempts to make an illegal access

(iii)   Attempts to execute code from a type of memory not permitted by the software environment[5]

(iv)   Attempts to write to EEPROM addresses containing data that should not be changed

(v)   Attempts to execute an illegal instruction code

(vi)   Attempts to alter registers controlling the operation of the TOE.

For the purposes of this threat, an illegal access is one that is marked "×" in the tables titled "Access" in the TOE memory map (section 3 of [HM]). In addition, among the accesses marked "○" in the tables, if a type of access allowed is noted under the circle, an access other than the allowed type of access is also an illegal access.

This threat applies whether the "attack" is deliberate or due to errors, but all the attacks covered are launched from software. Inducing errors by external means, as covered in T.Phys-Manipulation, may also give rise to the same sort of error conditions as listed for T.NoSWDetect.

T.NoSWResponse   Inability of Smartcard Embedded Software to respond to an attack

If Smartcard Embedded Software cannot detect a potential attack, or other dangerous condition, and has no ability to take action when such a condition is detected then there is a danger that it will not be able to prevent the attack continuing.

This threat does not address the particular details of individual attacks, but recognises that Smartcard Embedded Software may make checks on its own state to enhance protection against a variety of attacks (including those aimed at inducing errors by software or external means). For such checks to be useful, there must also be ways for the software to respond to the attack (e.g. by preventing further processing).

---

[5] The "software environment" is a term used to capture the definition of acceptable memory use for the software system using the TOE. This would usually be at the level of operating system software.

3.4   Organisational Security Policies

3.4.1   Policy Requirement from [BSI-PP-0002]

The following policy requirement is taken from section 3.4 of [BSI-PP-0002].

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 up to TOE Delivery) 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.

Assets relating specifically to P.Process-TOE are given in section 3.1 of [BSI-PP-0002].

Renesas implement the security measures to satisfy this policy requirement, and these are assessed as part of evaluation and certification against this ST. However, since they are not directly relevant to users of the TOE, the detailed measures and processes that implement the policy are not given here.

Note that the inclusion of identification information in EEPROM is described in more detail in section 2.4.2. This part of the policy establishes a basis for evaluation and security of software running on the chip, by ensuring that a TOE can be identified. Note that procedural measures (including Renesas' secure delivery procedures) will generally be required to ensure that TOE ICs are genuine, unless the Smartcard Embedded Software contains functionality to authenticate the IC[6].

P.Process-TOE covers identification of hard-coded Embedded Software (via identification of the ROM mask); soft-coded Embedded Software will generally need to provide its own identification.

3.4.2   Policy Requirement from [PA]

As an additional policy, the TOE provides specific security functionality which can be used by the Smartcard Embedded Software for cryptographic algorithm implementation. The policy P.Add-Functions is therefore adopted from [PA]. In the following policy, specific security functionality is listed which is not derived from threats identified for the TOE's environment

---

[6] For example, a hash or digital signature over a known area of memory might be provided by software.

because it can only be decided in the context of the smartcard application, against which threats the Smartcard Embedded Software will use the specific security functionality.

P.Add-Functions   Additional Specific Security Functionality

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

- Scrambled copy / compare / XOR to allow comparison, copy or XOR of data blocks containing secret information without exposing any of the data operated on to analysis via inherent leakage.

- Secure multiplication, to allow multiplication of data blocks containing secret information without exposing any of the data operated on to analysis via inherent leakage.

- Secure Modular Inversion function to securely calculate the public exponents of a key pair from the secret exponents, without exposing any of the data operated on to analysis via inherent leakage.

- Secure implementation of the 3DES algorithm using the hardware DES engine.

- Secure implementation of the standard RSA and RSA CRT algorithm (private and public key operations),

- Secure implementation of key generation for standard RSA and RSA CRT key pairs,

- Secure implementation of the hash functions SHA-1, SHA-256 and RipeMD-160.

4. Security Objectives

4.1 TOE Security Objectives

4.1.1 Objectives from [BSI-PP-0002]

The TOE shares the following high-level security considerations from section 4.1 of [BSI-PP-0002]:

SG1 maintain the integrity of User Data and of the Smartcard Embedded Software (when being executed/processed and when being stored in the TOE's memories).

SG2 maintain the confidentiality of User Data and of the Smartcard Embedded Software (when being processed and when being stored in the TOE's memories).

SG3 provide random numbers.

These high-level security considerations are refined below by defining security objectives as required by the Common Criteria. Note that maintaining the integrity of the TOE is a means to achieve these objectives.

O.Leak-Inherent Protection against Inherent Information Leakage

The TOE must provide protection against disclosure of confidential data (User Data or TSF Data) stored and/or processed in the smartcard IC by measurement and analysis of

- the shape and amplitude of signals (for example on the power, clock, or I/O lines),

- the time between events found by measuring signals (for example 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.

Note that this objective relates to security features provided by the TOE itself, and Smartcard Embedded Software should ensure that the security features are appropriately used in conjunction with any additional leakage countermeasures implemented in software (cf. A.Plat-Appl and A.Resp-Appl in section 3.2.1).

O.Phys-Probing Protection against Physical Probing

The TOE must provide protection against disclosure of User Data, against the disclosure/reconstruction of the Smartcard Embedded Software, and against the disclosure of other critical operational information. This includes protection against

- measuring through galvanic contacts, which is direct physical probing on the chip's surface other than on pads being bonded (using standard tools for measuring voltage and current),

- 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).

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.Phys-Probing assumes that the attacker has first performed reverse-engineering to understand the design and its properties and functions.

O.Malfunction   Protection against Malfunctions

The TOE must ensure its correct operation.

The TOE must prevent itself from being operated outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent errors. The environmental conditions may include voltage, clock frequency, temperature, or external energy fields.

An attacker may also attempt to cause the chip to malfunction by using a direct galvanic contact on elements on the chip surface.   This is considered as being a manipulation (see 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 physical manipulation of the TOE (including its software and TSF data), the Smartcard Embedded Software and User Data. This includes protection against

- reverse-engineering (understanding the design and its properties and functions),

- manipulation of the hardware and any data,

- controlled manipulation of memory contents (User Data).

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.Leak-Forced   Protection against Forced Information Leakage

The Smartcard must be protected against disclosure of confidential data (User Data or TSF data) processed in the Card (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 (cf. O.Malfunction) and/or

- by a physical manipulation (cf. O.Phys-Manipulation).

This objective includes resistance to attacks where T.Phys-Manipulation and T.Leak-Inherent are combined.

O.Abuse-Func   Protection against Abuse of Functionality

The TOE must prevent abuse of IC Dedicated Test Software (which may not be used after TOE Delivery) that would

(i)   disclose critical User Data,

(ii)   manipulate critical User Data of the Smartcard Embedded Software,

(iii)   manipulate Soft-coded Smartcard Embedded Software,

(iv)   bypass, deactivate, change or explore security features or functions of the TOE.

O.Identification   TOE Identification

The TOE must provide a 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.

The TOE identification data is described in section 2.4.2.

O.RND   Random Numbers

The TOE will ensure the cryptographic quality of random number generation. For instance, random numbers shall not be predictable and shall have 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.RND assumes that random number includes both true random number and pseudo random number.

4.1.2   Objectives Based on [PA]

This section includes an objective for the TOE to provide additional security functions, which are based on O.Add-Functions in [PA].

O.Add-Functions   Additional Specific Security Functionality

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

- Scrambled copy / compare / XOR to allow comparison, copy or XOR of data blocks containing secret information without exposing any of the data operated on to analysis via inherent leakage.

- Secure multiplication, to allow multiplication of data blocks containing secret information without exposing any of the data operated on to analysis via inherent leakage.

- Secure Modular Inversion function to securely calculate the public exponents of a key pair from the secret exponents, without exposing any of the data operated on to analysis via inherent leakage.

- Secure implementation of the 3DES algorithm using the hardware DES engine.

- Secure implementation of the standard RSA and RSA CRT algorithm (private and public key operations),

- Secure implementation of key generation for standard RSA and RSA CRT key pairs,

- Secure implementation of the hash functions SHA-1, SHA-256 and RipeMD-160

Note that to achieve an adequate strength of function for an application context, Smartcard Embedded Software may require use of triple-DES. The TOE Hardware enables this using repeated application of the DES coprocessor – this is discussed in section 14.3.1 of [HM] and in section 5.1.2.1. The triple-DES implementation realised in the TOE Software (ACL) uses the repeated application of the DES coprocessor in this fashion.

4.1.3   Other Objectives

The TOE offers additional facilities to protect embedded software from corrupted, erroneous, or malicious software. The same facilities may protect from some results of attempts at physical manipulation (cf. O.Phys-Manipulation).

A further high-level security consideration is therefore derived from SC4 in section 3.3.2.

SG4   software should be given the ability to detect and respond to attacks.

This leads to objectives that

O.SWDetect   Detection of potential attacks by the TOE

The following conditions shall be detected as an aid to identifying potential attacks:

(i)   Attempts to access memory outside an area defined for the software being executed

(ii)   Attempts to make an illegal access

   (iii) Attempts to execute code from a type of memory not permitted by the software environment[7]

   (iv) Attempts to write to EEPROM addresses containing data that should not be changed

   (v) Attempts to execute an illegal instruction code

   (vi) Attempts to alter registers controlling the operation of the TOE.

The Smartcard Embedded Software itself will determine the memory area for (i), the types of memory permitted for (iii), and the protected EEPROM addressed for (iv).

O.SWResponse   Response to potential attacks by Smartcard Embedded Software

                The TOE shall allow Smartcard Embedded Software to:

                (i) periodically interrupt processing to execute a known piece of Smartcard Embedded Software

                (ii) cause the processor to enter the reset state.

Executing a known piece of embedded software periodically gives embedded software the ability to check the execution state for any conditions that it wishes to monitor (in addition to those indicated under O.SWDetect above) and stop execution, or take some other action, if it detects a potential attack or other dangerous condition.

4.2   Environment Security Objectives

4.2.1   Environment Security Objectives from [BSI-PP-0002]

Phase 1

OE.Plat-Appl   Usage of Hardware Platform

                The Smartcard Embedded Software shall be designed so that the requirements from the following documents are met:

                (i) The TOE hardware manual [HM], user guidance manual [UGM], ACL User Manual [ACLSM], and the TOE application notes.

                (ii) Findings of the TOE evaluation reports relevant to the Smartcard Embedded Software.

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:

---

[7] The "software environment" is a term used to capture the definition of acceptable memory use for the software system using the TOE. This would usually be at the level of operating system software.

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 Smartcard Embedded Software as required by the security needs of the specific application context.

Because the TOE implements additional specific security functionality (as in O.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 keys must be unique with a very high probability, as well as cryptographically strong. For example, it must be ensured that it is not possible to derive the private key from a public key if asymmetric algorithms are used. If keys are imported into the TOE and/or derived from other keys, quality and confidentiality must be maintained. This implies that appropriate key management has to be realised in the environment.

From Phase 2 until TOE Delivery

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 and 3 up to TOE Delivery) 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 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. In order to make this practical, electronic identification shall be possible.

Assets relating specifically to the Organisational Security Policy P.Process-TOE (which is the source of this objective) are given in section 3.1 of [BSI-PP-0002].

From TOE Delivery until the beginning of Phase 7

OE.Process-Card   Protection during Packaging, Finishing and Personalisation

Security procedures shall be used after TOE Delivery up to delivery to the end-user 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 Phases after TOE Delivery up to the end of Phase 6 must be protected appropriately.

Assets relating specifically to the assumption A.Process-Card (which is the source of this objective) are given in section 3.1 of [BSI-PP-0002].

The precise nature of the protection required will depend on the application context.

4.2.2   Other Environment Security Objectives

For injected/pre-personalisation data, the sources and holders of the data need to support its security requirements.

OE.InjDatSupp   Injected Data Support

All data for injections/pre-personalisation shall be generated, distributed, maintained and destroyed in an adequately secure fashion. In general, the data shall be protected for both confidentiality and integrity.

Renesas ensures a secure interface with suppliers of this data by using the injection approach in section 2.4.2. Transmission of data to Renesas is secured by a variety of measures dependent on the transmission medium (e.g. ROM data may be sent by encrypted e-mail). The data is securely stored within the Renesas environment according to the medium.

5. IT Security Requirements

5.1 TOE Security Requirements

The functional requirements for the TOE come from 3 sources:

- [BSI-PP-0002]   These SFRs cover the core smart card IC requirements:

    - operating state – FRU_FLT.2 describes the usual operating conditions and requires the TOE to maintain a secure state; FPT_FLS.1 requires a secure response to exceptional operating conditions; FPT_SEP.1 requires the TOE to ensure that the secure responses to operating conditions are independent of software (since, for example, some software could be malicious)

    - controls over IC Dedicated Test Software – FMT_LIM.1 and FMT_LIM.2 require that the use of IC Dedicated Test Software for manufacturing tests must not allow security to be compromised after TOE Delivery

    - storage of identification/pre-personalisation data – FAU_SAS.1 requires that the TOE be capable of storing such data

    - protection against physical attacks – FPT_PHP.3 requires the TOE to be protected against physical tampering attacks

    - Protection against leakage – FDP_ITT.1 and FPT_ITT.1 require protection of data against leakage as it moves between components on the IC, and FDP_IFC.1 provides the policy of protection for data

    - Random number generation – FCS_RND.1 requires generation of good quality random numbers

- [PA]   This SFR covers cryptographic requirements.

    - 3DES – FCS_COP.1

    - RSA – FCS_COP.1 and FCS_CKM.1 are iterated for RSA and RSA CRT cryptographic operation, and key generation.

    - SHA-1 – FCS_COP.1

    - SHA-256 - FCS_COP.1

    - RipeMD-160 – FCS_COP.1

- TOE features   Some SFRs introduced from [BSI-PP-0002] are given a wider scope to reflect additional security features of the TOE and functions which support secure features in embedded software:

- Additional failure detection – the scope of FPT_FLS.1 includes additional software failure conditions trapped by the TOE support to embedded software to run tests at certain points during execution (enabling measures such as state integrity tests) and the ability of embedded software to cause a hardware reset (this is covered under FPT_FLS.1 in section 5.1.1.1).

- Controlled writes – FDP_ACC.1 [CRP] and FDP_ACF.1 [CRP] are added to represent the way in which some registers are protected against changes other than from embedded software executing in ROM; FDP_ACC.1 [WPP] and FDP_ACF.1 [CRP] specify the ability to prevent writing to chosen EEPROM pages.

Note that all SFRs are drawn from [CC/2] except for FCS_RND.1, FMT_LIM.1 & 2, and FAU_SAS.1, which are all defined in section 8 of [BSI-PP-0002].

5.1.1 TOE Security Functional Requirements from [BSI-PP-0002]

In the specifications of SFRs listed below, 'Refinement' sections are taken from [BSI-PP-0002]; 'Application Notes' add information specific to the TOE.

5.1.1.1 Operating Conditions

The TOE implements a pair of security functional requirements that ensure it operates within conditions under which it can maintain a secure state. This is represented in [BSI-PP-0002, fig 14], reproduced below:
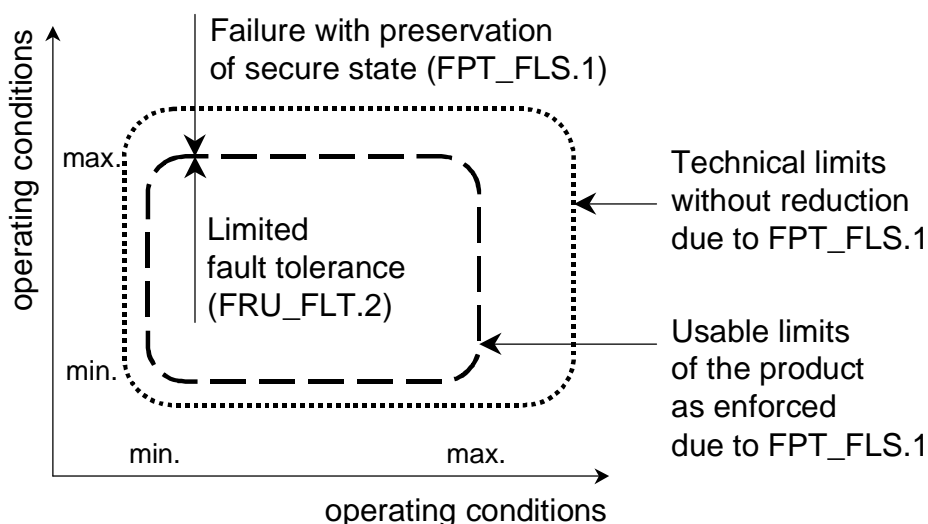

Figure 5-1: Paradigm Regarding Operating Conditions

Erroneous software conditions (some of which could arise from corruptions due to operating conditions) are dealt with under FPT_FLS.1.

FRU_FLT.2	Limited fault tolerance

Hierarchical to: FRU_FLT.1

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).

Dependencies: FPT_FLS.1 Failure with preservation of secure state

Refinement: The term "failure" above means "circumstances". The TOE prevents failures for the "circumstances" defined above.

FRU_FLT.2 thus states the condition for normal secure operation of the TOE functions (including coprocessors) within its expected operating conditions.

FPT_FLS.1	Failure with preservation of secure state

Hierarchical to: No other components

FPT_FLS.1.1   The TSF shall preserve a secure state when the following types of failures occur: exposure to operating conditions which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2) and where therefore a malfunction could occur.

Dependencies: ADV_SPM.1 Informal TOE security policy model

Refinement 1:   The term "failure" above also covers "circumstances". Then the TOE prevents failures for the "circumstances" defined above.

The refinement above is defined in [BSI-PP-0002]. An additional refinement is made here to capture features specific to the TOE.

Refinement 2:   The term "failure" above also covers the following:

(i)   Attempts to access memory outside an area defined for the software being executed

(ii)   Attempts to make an illegal access

(iii) Attempts to execute code from a type of memory not permitted by the software environment[8]

(iv)   Attempts to execute an illegal instruction code

(v)   Processor halted by Smartcard Embedded Software

---

[5] The "software environment" is a term used to capture the definition of acceptable memory use for the software system using the TOE. This would usually be at the level of operating system software.

Application notes:

1. The TOE is required to implement a periodic interrupt, and an interrupt whenever a write to EEPROM takes place, to enable detection of failure conditions and preservation of a secure state by Smartcard Embedded Software.

FPT_SEP.1   **TSF domain separation**

Hierarchical to:	No other components
FPT_SEP.1.1	The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.
FPT_SEP.1.2	The TSF shall enforce separation between the security domains of subjects in the TSC.
Dependencies:	No dependencies.
Refinement:	The components which support the SFRs "Limited fault tolerance (FRU_FLT.2)" and "Failure with preservation of secure state (FPT_FLS.1)" shall be protected from interference of the Smartcard Embedded Software.

5.1.1.2 Protection Against Abuse of Functionality

The TOE controls access to test mode. Following [BSI-PP-0002], this is specified using the extended functional family FMT_LIM, defined in section 8.5 of [BSI-PP-0002].

FMT_LIM.1   **Limited capabilities**

Hierarchical to:	No other components
FMT_LIM.1.1	The TSF shall be designed in a manner that limits their capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: Deploying Test Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks.
Dependencies:	FMT_LIM.2 Limited availability.

Application notes:

1. The "capabilities" referred to in FMT_LIM.1 are the functions implemented in the IC Dedicated Test Software.

FMT_LIM.2   **Limited availability**

Hierarchical to:	No other components
FMT_LIM.2.1	The TSF shall be designed in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the

following policy is enforced: Deploying Test Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks.

Dependencies: FMT_LIM.1 Limited capabilities.

Application notes:

1. The "capabilities" referred to in FMT_LIM.2 are the functions implemented in the IC Dedicated Test Software.

5.1.1.3   Storage of Production Data

The TOE stores identification/pre-personalisation data as described in section 2.4.2. This is included in [BSI-PP-0002] as the CC Part 2 extended functional component FAU_SAS.1, replacing FAU_GEN.1, and defined in section 8.6 of [BSI-PP-0002].

FAU_SAS.1   Audit storage

Hierarchical to: No other components

FAU_SAS.1.1   The TSF shall provide test personnel before TOE Delivery with the capability to store the Initialisation Data and/or Pre-personalisation Data and/or supplements of the Smartcard Embedded Software in the audit records.

Dependencies: No dependencies

Application notes:

1. The data covered by "Initialisation Data and/or Pre-personalisation Data and/or supplements of the Smartcard Embedded Software" is as identified in section 2.4.2.

The function Audit storage (FAU_SAS.1) is subject to the limitations as specified by Limited availability (FMT_LIM.2) above.

5.1.1.4   Protection against Physical Manipulation and Probing

FPT_PHP.3   Resistance to physical attack

Hierarchical to: No other components

FPT_PHP.3.1   The TSF shall resist physical manipulation and physical probing to the TSF by responding automatically such that the TSP is not violated.

Dependencies: No dependencies

Refinement: The TOE will implement appropriate measures to continuously counter physical manipulation and physical probing. Due to the nature of these attacks (especially manipulation) the TOE can by no

means detect attacks on all of its elements. Therefore, permanent protection against these attacks is required ensuring that the TSP could not be violated at any time. Hence, "automatic response" means here (i) assuming that there might be an attack at any time and (ii) countermeasures are provided at any time.

5.1.1.5   Protection against Leakage

The following SFRs are required as part of protection against inherent leakage. FDP_ITT.1, FPT_ITT.1, and FDP_IFC.1 are iterated here to address protection and control implemented by the hardware and by the ACL. For brevity, they are only listed in detail once below. Please see the Application Notes at the end of this section for details of the separation.

FDP_ITT.1	Basic internal transfer protection

FDP_ITT.1 [HW][ACL]

Hierarchical to:	No other components
FDP_ITT.1.1 [HW][ACL]	The TSF shall enforce the Data Processing Policy to prevent the disclosure of user data when it is transmitted between physically-separated parts of the TOE.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
Refinement:	The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as physically-separated parts of the TOE.

The Data Processing Policy is defined under FDP_IFC.1 below.

FPT_ITT.1	Basic internal TSF data transfer protection

FPT_ITT.1.1 [HW][ACL]

Hierarchical to:	No other components
FPT_ITT.1.1 [HW][ACL]	The TSF shall protect TSF data from disclosure when it is transmitted between separate parts of the TOE.
Dependencies:	No dependencies.
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.

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. |

FDP_IFC.1   **Subset information flow control**

FDP_IFC.1.1 [HW][ACL]

Hierarchical to: No other components

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

Dependencies: FDP_IFF.1 Simple security attributes

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

Application note:

1. The above three TSFs are iterated for the TOE hardware that is required to provide specific measures to implement them as: FDP_ITT.1 [HW], FPT_ITT.1 [HW] and FDP_IFC.1 [HW].

2. The above three TSFs are iterated for the ACL (TOE Software), which is required to provide specific measures to protect information using: scrambled data operations to copy / compare and XOR data; and operand and exponent blinding for functions implemented on the MMC, including secure multiply and secure modular inversion. They are iterated as: FDP_ITT.1 [ACL], FPT_ITT.1 [ACL] and FDP_IFC.1 [ACL].

5.1.1.6   Cryptographic Support

The TOE generates random numbers that can be used for cryptographic key generation. To capture the functional requirement, the family FCS_RND, defined in section 8.4 of [BSI-PP-0002] is used.

FCS_RND.1   **Quality metric for random numbers**

Note that the TOE includes dedicated hardware for true random number generation (along with interface functions in the ACL to access and support this hardware) along with a software implementation of a pseudo random number generator. Therefore this family is iterated twice, for True Random Number Generator (TRNG) and Pseudo Random Number Generator (PRNG) respectively.

True Random Number Generator

FCS_RND.1 [TRNG]

FCS_RND.1.1 [TRNG]   The TSF shall provide a mechanism to generate random numbers that meet the requirements of the monobit, poker, runs, long run, and autocorrelation tests in [AIS31].

<u>Pseudo Random Number Generator</u>

FCS_RND.1 [PRNG]

FCS_RND.1.1 [PRNG]   The TSF shall provide a mechanism to generate random numbers that meet the requirements of the formal and statistical tests to meet functionality class K3 as defined in [AIS20].

5.1.2   TOE Security Functional Requirements Based on [PA]

5.1.2.1   Cryptographic Support

The TOE includes a hardware implementations of a DES coprocessor and an MMC (Modular Multiplication Coprocessor), and software implementations of 3DES, RSA and hashing functions (which are implemented using the hardware DES and MMC as appropriate).

FCS_COP.1   Cryptographic operation

FCS_COP.1 is iterated here to address the cryptographic functions implemented by the TOE.

<u>DES Encryption and Decryption</u>

FCS_COP.1 [3DES]

Hierarchical to: No other components

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

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

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

Application notes:

1. Although the TOE provides a DES coprocessor, DES is only ever used as an encryption function in the context of particular Smartcard Embedded Software, and for this reason it is noted that application software may need to use triple DES to achieve a suitable strength. In this case, Smartcard Embedded Software shall use the TOE to implement triple DES, as described in the guidance for software developers in section 14.3.1 of [HM], or may use the triple DES function of the ACL component.

2. Although Smartcard Embedded software may use the DES coprocessor directly it is recommended to use it through the ACL functions.

RSA Encryption and Decryption

FCS_COP.1 [RSA]

Hierarchical to:	no other components

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

ISO/IEC 9796-2, Annex A

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

RSA Key Generation

FCS_CKM.1 [RSA]

Hierarchical to:	No other components

FCS_CKM.1.1 [RSA]   The TSF shall generate cryptographic keys in a accordance with a specified cryptographic key generation algorithm Rivest-Shamir-Adleman (RSA) and specified cryptographic key sizes 512-2048 bit that meet the following list of standards:

ISO/IEC 9796-2, Annex A

Dependencies: [FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

Secure Hash Functions

FCS_COP.1 [SHA-1]

Hierarchical to:	no other component

FCS_COP.1.1 [SHA-1]   The TSF shall perform hashing in accordance with a specified cryptographic algorithm Secure Hash Algorithm (SHA-1) and cryptographic key sizes [assignment: cryptographic key sizes] that meets the following standards:

Secure Hash Standard, Federal Information Processing Standards Publication 180-2, 2002

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

FMT_MSA.2 Secure security attributes

Application Notes:

1. Hash operations do not use a cryptographic key; the assignment of keysize is not applicable and thus not fulfilled.

FCS_COP.1 [SHA-256]

Hierarchical to:	no other component
FCS_COP.1.1 [SHA-256]	The TSF shall perform hashing in accordance with a specified cryptographic algorithm Secure Hash Algorithm (SHA-256) and cryptographic key sizes [assignment: cryptographic key sizes] that meets the following standards:
	Secure Hash Standard, Federal Information Processing Standards Publication 180-2, 2002
Dependencies:	[FDP_ITC.1 Import of user data without security attributes or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes

Application Notes:

1   Hash operations do not use a cryptographic key; the assignment of keysize is not applicable and thus not fulfilled.

FCS_COP.1 [RIPEMD-160]

Hierarchical to:	no other component
FCS_COP.1.1 [RIPEMD-160]	
	The TSF shall perform hashing in accordance with a specified cryptographic algorithm RIPEMD-160 and cryptographic key sizes [assignment: cryptographic key sizes] that meets the following standards:
	ISO/IEC FDIS 10118-3:2003 (Information technology - Security techniques - Hash-functions - Part3)
Dependencies:	[FDP_ITC.1 Import of user data without security attributes or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes

Application Notes:

1. Hash operations do not use a cryptographic key; the assignment of keysize is not applicable and thus not fulfilled.

5.1.3 Other TOE Security Functional Requirements

FDP_ACC.1 Subset access control

FDP_ACC.1 is iterated here to address both the control over software to write to certain controlled registers (FDP_ACC.1 [CRP]) and software to write to protected pages of EEPROM memory (FDP_ACC.1 [WPP]).

FDP_ACC.1 [CRP]

Hierarchical to:	No other components

FDP_ACC.1.1 [CRP]   The TSF shall enforce the Controlled-Register Policy on software to set bits in defined, security-relevant registers.

Here,
policy: Controlled-Register Policy
Subjects: software
Objects: bits in defined, security-relevant registers
Operations: to set

Controlled-Register Policy

The ability to set the values in the controlled registers is restricted to software instructions executed in ROM.

Dependencies: FDP_ACF.1 Security attribute based access control

Application Notes:

1. The controlled registers are those defined in FDP_ACC.1.1 [CRP].

FDP_ACC.1 [WPP]

Hierarchical to:	No other components

FDP_ACC.1.1 [WPP]   The TSF shall enforce the Write-Protect Policy on software to write to EEPROM.

Here,
policy: Write-Protect Policy
Subjects: software
Objects: EEPROM
Operations: to write

Write-Protect Policy   An attempt to write to EEPROM shall fail, leaving the previous memory contents unaltered if the relevant EEPROM page is write protected.

Once a page is write-protected, this protection cannot be removed.

Dependencies: FDP_ACF.1 Security attribute based access control

Application Notes:

1. The setting of write protection is described in [HM].

2. All modes of EEPROM write (see [HM]) are covered by this policy.

FDP_ACF.1   Security attribute based access control

FDP_ACF.1 is iterated here to address both the security attribute based access control over software to write to certain controlled registers (FDP_ACF.1 [CRP]) and software to write to protected pages of EEPROM memory (FDP_ACF.1 [WPP]).

FDP_ACF.1 [CRP]

Hierarchical to: No other components

FDP_ACF.1.1 [CRP][9]   The TSF shall enforce the Controlled-Register Policy to objects based on the following:

                              Subject: software running on the ROM
                              Object: control registers relating to the FMU
                              attributes: to have read and/or write access normally

FDP_ACF.1.2 [CRP]   The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: evaluate the corresponding permission control information: "The ability to set the values in the controlled registers is restricted to software operations executed from the ROM." before, during or after the access so that accesses to be denied can not be utilised by the subject attempting to perform the operation.

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

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

Dependencies: FDP_ACC.1 Subset access control
                             FMT_MSA.3 Static attribute initialisation

---

[9] FDP_ACF.1.1 is changed as follows according to Final Interpretation 103
    FDP_ACF.1.1 The TSF shall enforce the [assignment: access control SFP] to objects based on the following: [assignment: list of subjects and objects controlled under the indicated SFP, and. for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes].

FDP_ACF.1 [WPP]

Hierarchical to:	No other components

FDP_ACF.1.1 [WPP][9]   The TSF shall enforce the Write Protect Policy to objects based on the memory area where the access is performed and the operation to be performed.

FDP_ACF.1.2 [WPP]   The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: evaluate the corresponding permission control information: "An attempt to write to EEPROM shall fail, leaving the previous memory contents unaltered if write protection is set for the relevant EEPROM page. Once a page is write-protected, this protection cannot be removed." before, during or after the access so that accesses to be denied can not be utilised by the subject attempting to perform the operation.

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

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

Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation

5.1.4 TOE Security Assurance Requirements

The assurance level for this Security Target is EAL4 augmented, as in section 5.1.2 of [BSI-PP-0002].

The augmentations to EAL4 are:

- ADV_IMP.2 - this adds the full implementation representation of the security functions to the evaluation scope

- ALC_DVS.2 - this increases the confidence in the vital area of developer security measures to the highest CC level

- AVA_MSU.3 - this adds evaluator testing of the potential for misconfiguration of the TOE within the evaluation scope

- AVA_VLA.4 - this increases the scope of vulnerability analysis and penetration testing to the highest CC level.

The minimum strength of function is SOF-high. In addition, the quality of the mechanisms contributing to the information leakage resistance of the DES coprocessor and ACL can be analysed using probabilistic methods based on measurement of the power consumption of the TOE - SOF-high is also claimed for this mechanism. Also note that the strength of the cryptographic algorithms is out of the scope of the TOE.

Refinements of the assurance requirements required by [BSI-PP-0002], and implemented by Renesas for the TOE, are described in section 5.1.3 of [BSI-PP-0002].

5.2   Security Requirements for the Environment

5.2.1   Security Requirements for the IT Environment from [PA]

In section 5.2.1 of [BSI-PP-0002], only non-IT security requirements are placed on the environment (see section 5.2.2). Although these requirements are applicable to Smartcard Embedded Software, it is not necessary or appropriate to place specific functional requirements.

However, the following requirements for the environment are adopted from [PA].

The security functional requirement "Cryptographic operation (FCS_COP.1)" met by the TOE has the following dependencies

-   [FDP_ITC.1 Import of user data without security attributes or FCS_CKM.1 Cryptographic key generation],

-   FCS_CKM.4 Cryptographic key destruction,

-   FMT_MSA.2 Secure security attributes.

These requirements all address the appropriate management of cryptographic keys used by the specified cryptographic function. The requirement "Cryptographic Key Generation" for RSA ("FCS_CKM.1 [RSA]"), is met by the TOE (implemented in the ACL). All other requirements concerning key management, including "Cryptographic key generation" for 3DES, 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. Note that, however, for FCS_COP.1, the dependencies are unfulfilled because the environment is unknown (it is an issue for the user of the chip of course).

The environment shall meet the requirement "Import of user data without security attributes (FDP_ITC.1)" or "Cryptographic key generation (FCS_CKM.1)" (except as previously noted for FCS_CKM.1 [RSA]) as specified below. Note that, however, for FDP_ITC.1, the operations are deliberately uncompleted and dependencies unfulfilled because the environment is unknown (it is an issue for the user of the chip of course). Note also that the secure hash functions specified by FCS_COP.1 [SHA-1], FCS_COP.1 [SHA-256] and FCS_COP.1 [RIPEMD-160] do not use cryptographic keys, so the FCS_CKM.1 dependency can be considered to be automatically discharged for these iterations of FCS_COP.1 without any special consideration. This means the environment need only fulfil the dependency FCS_CKM.1 for the 3DES iterations of FCS_COP.1.

FDP_ITC.1   Import of user data without security attributes

Hierarchical to: No other components.

FDP_ITC.1.1   The TSF shall enforce the [assignment: access control SFP and/or information flow control SFP] when importing user data, controlled under the SFP, from outside of the TSC.

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

FDP_ITC.1.3   The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [assignment: additional importation control rules].

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

FMT_MSA.3 Static attribute initialisation

FCS_CKM.1   Cryptographic key generation

Hierarchical to: No other components.

FCS_CKM.1.1   The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: cryptographic key generation algorithm] and specified cryptographic key sizes [assignment: cryptographic key sizes], that meet the following: [assignment: list of standards].

Dependencies: [FCS_CKM.2 Cryptographic key distribution or FCS_COP.1 Cryptographic operation]

FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

Note the TOE itself meets requirement FCS_CKM.1 [RSA] by use of the ACL RSA Cryptographic key generation functions.

The environment shall meet the requirement "Cryptographic key destruction (FCS_CKM.4)" as specified below.   Note that, however, for FCS_CKM.4, the operations are deliberately uncompleted and dependencies unfulfilled because the environment is unknown (it is an issue for the user of the chip of course).   Note also that the secure hash functions specified by FCS_COP.1 [SHA-1], FCS_COP.1 [SHA-256] and FCS_COP.1 [RIPEMD-160] do not use cryptographic keys so the FCS_CKM.4 dependency can be considered to be automatically discharged for these iterations of FCS_COP.1 without requiring any special consideration.   Therefore the environment need only fulfil the FCS_CKM.1 dependency for the RSA and 3DES iterations of FCS_COP.1.

FCS_CKM.4   Cryptographic key destruction

Hierarchical to: No other components.

FCS_CKM.4.1   The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: cryptographic key destruction method] that meets the following: [assignment: list of standards].

Dependencies: [FDP_ITC.1 Import of user data without security attributes or FCS_CKM.1 Cryptographic key generation]

FMT_MSA.2 Secure security attributes

The environment shall meet the requirement "Secure security attributes (FMT_MSA.2)" as specified below. Note that, however, for FMT_MSA.2, the operations are deliberately uncompleted and dependencies unfulfilled because the environment is unknown (it is an issue for the user of the chip of course).

FMT_MSA.2   Secure security attributes

Hierarchical to: No other components.

FMT_MSA.2.1   The TSF shall ensure that only secure values are accepted for security attributes.

Dependencies: ADV_SPM.1 Informal TOE security policy model

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

FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

5.2.2   Security Requirements for the Non-IT Environment

5.2.2.1   Security Requirements for the Non-IT Environment from [BSI-PP-0002]

As in section 5.2.2 of [BSI-PP-0002], this ST applies the following requirements:

RE.Phase-1   Design and Implementation of the Smartcard Embedded Software

The developers shall design and implement the Smartcard Embedded Software in such way that it meets the requirements from the following documents:

(i) The TOE hardware manual [HM], user guidance manual [UGM], ACL Software User Manual [ACLSM], and the hardware application notes.

(ii) Major findings of the hardware evaluation reports relevant for the Smartcard Embedded Software.

The developers shall implement the Smartcard Embedded Software in a way that it protects security relevant User Data (especially cryptographic keys) as required by the security needs of the specific application context.

In particular, the Smartcard Embedded Software shall not disclose secret User Data to unauthorised users or processes as defined for the application context. Similarly the Smartcard Embedded Software shall not allow

unauthorised users or processes to use or modify security relevant User Data.

RE.Process-Card   **Protection during Packaging, Finishing and Personalisation**

The Card Manufacturer (after TOE Delivery up to the end of Phase 6) shall use adequate security measures 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).

Section 8.2.2 of [BSI-PP-0002] gives examples of ways in which Smartcard Embedded Software may implement security issues. The TOE does not itself require any of these functions to be implemented in software – this is entirely a decision to be based on the software context. However, the TOE aims to provide support for software implementation of FDP_SDI.1 and FPT_AMT.1 (the examples given in [BSI-PP-0002]) through features such as the EWE interrupt – cf. FPT_FLS.1 in section 5.1.1.1 and SF.ESFunctions in section 6.1.

5.2.2.2   Security Requirements for the Non-IT Environment from [PA]

For implementation of cryptographic functions in Smartcard Embedded Software, the following requirement from [PA] is included:

RE.Cipher   Cipher Schemes

The developers of Smartcard Embedded Software must not implement routines in a way which may compromise keys when the routines are executed as part of the Smartcard Embedded Software. Performing functions which access cryptographic keys could allow an attacker to misuse these functions to gather information about the key which is used in the computation of the function.

Keys must be kept confidential as soon as they are generated. The keys must be unique with a very high probability, as well as cryptographically strong. For example, it must be ensured that it is not possible to derive the private key from a public key if asymmetric algorithms are used. If keys are imported into the TOE and/or derived from other keys quality and confidentiality must be maintained. This implies that an appropriate key management has to be realised in the environment.

6.   TOE Summary Specification

6.1   TOE Security Functions

1.   SF.HWProtect

The TOE is protected from attacks on the operation of the IC hardware. The protection features include detection of out-of-range supply voltages, frequencies or temperatures, detection of illegal address or instruction, and physical security.

Detection of an error causes the AE55C1 to enter a reset state.

Correct operating ranges are defined in [HM].

SF.HWProtect uses probabilistic or permutational effects for the memory address and data encryption and therefore has to be included in the AVA_SOF analysis with SOF-high.

2.   SF.LeakProtect

The TOE Hardware protects against leakage of information from the IC. The protection features include:

- Functions designed to alter the power consumption of the device

- DES protection – the DES coprocessor contains additional measures to resist side-channel attacks.

SF.LeakProtect uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF-high.

3.   SF.ACL-LeakProtect

In addition to the measures provided by the TOE Hardware, the ACL provides additional measures to protect against leakage of information from the IC. The protection features include:

- Scrambled copy / compare / XOR – the ACL provides functions for securely copying, comparing or XORing strings of bytes.

- Secure multiply – the ACL provides a function to securely multiply two large numbers using the MMC.

- Secure Modular Inversion – the ACL provides a function to securely calculate the modular inversion of a big number.

- Secure RSA operation and keyset generation – the ACL uses its own secure copy/compare and secure multiply in order to realise the RSA (CRT and standard modular exponentiation, and also keyset generation) functions.

SF.ACL-LeakProtect uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF-high.

4. SF.RNG

The TOE includes a physical random number generator (HW RNG) designed to produce random numbers for the generation of cryptographic keys and for other critical uses. This random number generator meets the requirements of application class P2 (as specified in [AIS31]).

The TOE can be used to generate 64-bit random numbers which satisfy the requirements of the monobit, poker, runs, long run, and autocorrelation tests in [AIS31].

RNG support function – The ACL includes a function to access the HW RNG, which automatically runs an on-line test of the HW RNG to verify that its operation has not been compromised. This function implements "Poker Test" according to [UGM].

Since generating random numbers using the HW RNG is time consuming, the TOE also includes a software implementations of a pseudo random number generator (PRNG). The PRNG is initialised with a seed from the HW RNG. This PRNG is implemented according to the standard ANSI X9.31 Appendix A (which replaces ANSI X9.17 Appendix C) and meets functionality class K3 (as specified in [AIS20]).

Application Notes

The TOE provides a tamper resistant hardware random number generator (see section 10 of [HM]). However, in order to provide additional assurance to the User software that the hardware is functioning, [AIS31] requires the use of an on-line test of the RNG. This online test must be used to verify the correct operation of the HW RNG in at least the following two cases:

1. At the start of each Smart Card Application session.

2. Immediately before an RSA keyset generation is started.

SF.RNG uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF-high.

5. SF.DES

The TOE provides a hardware DES coprocessor that carries out DES encryption and decryption in ECB mode; and ACL software functions to access this coprocessor for DES and to implement Triple-DES, according to the following standard:

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

Application Notes

1. The DES coprocessor implements single DES encrypt and decrypt operations, and has been optimised to allow easy implementation of triple-DES (as described in the standard) in Smartcard Embedded Software, as described in section 14.3.1 of [HM].

2. The ACL software function for triple-DES is implemented using the specific features of the DES coprocessor, as described in section 14.3.1 of [HM].

3. Although the TOE provides a DES coprocessor, and ACL functions to access and implement DES and Triple-DES respectively, only Triple-DES has sufficient SOF to be claimed as a Security Function and for use in secure part of any Smartcard Embedded Software.

4. The TOE implements ECB mode - software can implement other modes such as CBC.

5. To provide secure embedded software, the software developer is required to ensure that the DES and (where used in a cryptographic algorithm) modular multiplication coprocessors, and related ACL functions, are used in a way that does not compromise the key or plain text (see A.Plat-Appl, A.Resp-Appl and A.Key-Function). Guidance for the implementation of secure Smartcard Embedded Software is given in [UGM] and [ACLSM].

DES support function – The ACL provides an interface function for the DES coprocessor, and in addition implements a triple DES function that uses the DES coprocessor in the fashion indicated in section 14.3.1 of [HM]. The DES and 3DES algorithms are implemented according to the standard U.S. Department of Commerce / National Bureau of Standards Data Encryption Standard (DES), FIPS PUB 46-3, 1999 October 25. Only triple-DES has sufficient SOF to be claimed as a security function and for use in the secure part of any Smartcard Embedded Software.

SF.DES uses probabilistic or permutational effects but does not need to be included in the AVA_SOF analysis with SOF-high since the function is beyond the scope of a Common Criteria evaluation.

6. SF.FMU

The TOE firewall management unit enables software to control addresses that can be accessed in the following two ways:

a) The FMU checks that a target address used in any instruction is within/out of specified limits.

If a target address is not within accessible areas then the TOE enters the reset state or FMU interrupt.

b) In addition, the FMU may enforce a policy that the TOE may not execute code in either EEPROM or RAM, or both.

Changes to these policies can only be made by software executing in ROM.

Application Notes:

1. The accessible address registers are defined in section 12.2 of [HM].

SF.FMU does not use probabilistic or permutational effects.

7. SF.ESFunctions

The scope of the TOE evaluation includes correct operation of aspects such as the CPU instructions, memory functions and standard peripherals such as memories, registers, I/O

interfaces, timers and UART as specified in [HM]. In addition, the TOE offers hardware and software facilities that are designed to enable Smartcard Embedded Software to address threats to its correct operation by taking control of the operating environment. The Smartcard Embedded Software developer can rely on the following TOE functionality that has been specifically evaluated as part of the TOE:

- EWE Interrupt

  Every time the TOE writes to EEPROM, it generates a non-maskable interrupt (the EWE interrupt). When this interrupt occurs, execution is passed to a user-definable address held in the EWE vector. A user can therefore add code at this location to carry out a variety of checks, for example to confirm the integrity of data, or the context in which certain areas of EEPROM are being written.

  If a new EWE interrupt is received before the previous one has been cleared then the TOE enters the reset state.

- CPU Halt

  When the halt bit is set by user software, the TOE will stop execution until an external reset is received.

SF.ESFunction does not use probabilistic or permutational effects.

8. SF.TestModeControl

Once the TOE has been set to user mode, test mode functions are no longer accessible.

SF.TestModeControl does not use probabilistic or permutational effects.

9. SF.EEPAccess

The TOE allows any page of EEPROM to have writes (or erase) disallowed by setting the page to have a protected state. If a write (or erase) is attempted to a protected page then it will leave the page content unaltered.

Protection of a page against writes is permanent once set.

SF.EEPAccess does not use probabilistic or permutational effects.

10. SF.Inject

During manufacture, each TOE is injected with data that uniquely identifies the individual IC. If specified for the Smartcard Embedded Software included, then additional data (some of it IC-specific) may also be injected during manufacture.

SF.Inject does not use probabilistic or permutational effects.

11. SF.MMCopro

The TOE provides a coprocessor that carries out modular multiplication according to the specification as in section 15 of [HM].

This forms the basis for software implementation of algorithms such as RSA.

Secure RSA Operation and Keyset Generation – The ACL provides an implementation of the RSA and RSA CRT algorithm using the hardware MMC. It also provides secure generation of keys for the RSA and RSA CRT algorithm using the hardware RNG and software PRNGs. These are implemented according to standard ISO/IEC 9796-2, Annex A.

The RSA and RSA CRT encryption, decryption and key generation functions of the ACL can handle key sizes ranging from 512 to 2048 bits in length, however only key sizes above and including 1024 bits in length are considered for the purpose of common criteria evaluation.

SF.MMCopro uses probabilistic or permutational effects but does not need to be included in the AVA_SOF analysis with SOF-high since the function is beyond the scope of a Common Criteria evaluation.

12. SF.Hash

Hash functions - The ACL provides implementations of the hash functions SHA-1, SHA-256 and RipeMD-160 according to the standards Secure Hash Standard, Federal Information Processing Standards Publication 180-2, 2002 and ISO/IEC FDIS 10118-3, 2003: Information Technology – Security techniques – Hash-functions – Part 3 respectively.

SF.Hash uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF-high.

The Reset State

The reset state is referred to in several of the Security Functions above. In the reset state, the chip stops execution until a reset signal is received on the reset line. When a reset occurs, the following actions are carried out:

- The CPU resets to its initial state

- Registers are set to their initial values (as defined in [HM])

- Execution begins at the location in the reset vector (as defined in [HM]).

6.2 Correspondence Between TOE Security Function and SFR

Table below shows the correspondence between SFRs and each of the Security Functions provided in the section 6.1 above.

Table 6-1: TOE Security Functions Mapping to SFRs

TOE Security Function	SFR
SF.HWProtect,	FRU_FLT.2, FPT_FLS.1, FPT_SEP.1, FPT_PHP.3
SF.LeakProtect	FDP_ITT.1 [HW], FPT_ITT.1 [HW], FDP_IFC.1 [HW]
SF.ACL-LeakProtect	FDP_ITT.1 [ACL], FPT_ITT.1 [ACL], FDP_IFC.1 [ACL]
SF.RNG	FCS_RND.1 [TRNG], FCS_RND.1 [PRNG]
SF.DES	FCS_COP.1 [3DES]
SF.FMU	FRU_FLT.2, FPT_FLS.1, FPT_SEP.1, FPT_PHP.3, FDP_ACC.1 [CRP], FDP_ACF.1 [CRP]
SF.ESFunctions,	FRU_FLT.2, FPT_FLS.1, FPT_SEP.1, FPT_PHP.3
SF.TestModeControl	FMT_LIM.1, FMT_LIM.2
SF.EEPAccess	FPT_FLS.1, FPT_SEP.1, FDP_ACC.1 [WPP], FDP_ACF.1 [WPP]
SF.Inject	FAU_SAS.1
SF.MMCopro	FCS_COP.1 [RSA], FCS_CKM.1 [RSA]
SF.Hash	FCS_COP.1 [SHA-1], FCS_COP.1 [SHA-256], FCS_COP.1 [RIPEMD-160]

6.3 Assurance Measures

The TOE meets the requirements for EAL4 (as defined in [CC/3]) and the following augmentations:

- ADV_IMP.2

- ALC_DVS.2

- AVA_MSU.3

- AVA_VLA.4

It also meets the requirements of the refinements made to the assurance requirements in section 5.1.3 of [BSI-PP-0002].

The table below maps assurance requirements to the documents describing the relevant requirements:

Table 6-2: Documents Meeting Assurance Requirements

Document	[CC/3] Evidence Item	Related Assurance Components
Security Policy Model	Security Policy Model	ADV_SPM
	Security Function	ADV_FSP
Hardware Manual	Functional Specification	ADV_FSP
	User Guidance and administrator guidance documentation	AGD_USR, AGD_ADM
User Guidance Manual ACL Manual	User Guidance and administrator guidance documentation	AGD_USR, AGD_ADM ADO_IGS
Correspondence Mapping	Correspondence analysis between TOE Summary Specification and functional specification, high level design, low level design and implementation	ADV_RCR
High-Level and Low-Level Design Documentation Set	High-level design, low-level design	ADV_HLD, ADV_LLD
Source Code*	Implementation	ADV_IMP
Test Documentation Set	Testing, test depth and test coverage documentation	ATE
Site Information	Development tools documentation	ALC
	Configuration management documentation	ACM
	Delivery procedures	ADO_DEL
Misuse Analysis	Misuse Analysis	AVA_MSU
Strength of Function Analysis	Strength of Function Analysis	AVA_SOF
Vulnerability Analysis	Vulnerability Analysis	AVA_VLA

Note: The source code exists in an electronic data form.

7.   PP Claims

7.1   PP Reference

This ST conforms to [BSI-PP-0002].

Note that [PA] is used to define additional requirements relating to cryptographic functions. This ST is also [PA] conformant.

7.2   PP Tailoring

FCS_RND.1 is completed with a quality metric – see section 5.1.1.6.

7.3   PP Additions

The inclusions from [BSI-PP-0002] are clearly shown in the relevant section titles. All other assumptions, threats, objectives and SFRs, in sections 3.2.2, 3.2.3, 3.3.2, 3.4.2, 4.1.2, 4.1.3, 4.2.2, 5.1.2, 5.1.3, 5.2.1, and 5.2.2.2 are additional to those in the PP.

8. Rationale

8.1   Security Objectives Rationale

The way in which [BSI-PP-0002] assumptions, organisational security policy and threats are met by objectives is given in section 7.1 of [BSI-PP-0002]. The table below includes the mapping from section 7.1 of [BSI-PP-0002] and adds the rationale for the additional assumptions, policy and threats in this Security Target.

Table 8-1: Coverage of Security Assumptions, Policies and Threats by Objectives

Assumption/Threat/ Organisational Security Policy	Addressed by Objective
A.Key-Function	OE.Plat-Appl, OE.Resp-Appl
A.Plat-Appl	OE.Plat-Appl
A.Process-Card	OE.Process-Card
A.Resp-Appl	OE.Resp-Appl
A.InjDatSupp	OE.InjDatSupp
P.Add-Functions	O.Add-Functions
P.Process-TOE	OE.Process-TOE, O.Identification
T.Leak-Inherent	O.Leak-Inherent
T.Phys-Probing	O.Phys-Probing
T.Phys-Manipulation	O.Phys-Manipulation
T.Malfunction	O.Malfunction
T.Leak-Forced	O.Leak-Forced
T.Abuse-Func	O.Abuse-Func
T.NoSWDetect	O.SWDetect
T.NoSWResponse	O.SWResponse
T.RND	O.RND

A.Key-Function is enforced by OE.Plat-Appl and OE.Resp-Appl, which directly requires the embedded software to use the features in TOE documentation to take measures to ensure that keys are not compromised by the way in which the TOE's cryptographic functions are used. Note that this recognises the fact that measures in hardware are only part of the solution for software TOEs, which must also ensure that their algorithms protect keys.

A.Plat-Appl is enforced by a directly corresponding requirement on the environment in OE.Plat-Appl. (Note that as in section 7.1 of [BSI-PP-0002] this applies to Phase 1 of the lifecycle.)

A.Process-Card is enforced by a directly corresponding requirement on the environment in OE.Process-Card. (Note that as in section 7.1 of [BSI-PP-0002] this applies to Phases 4-6 of the lifecycle.)

A.Resp-Appl is enforced by a directly corresponding requirement on the environment in OE.Resp-Appl. (Note that as in section 7.1 of [BSI-PP-0002] this applies to Phase 1 of the lifecycle.)

A.InjDatSupp is enforced by a directly corresponding requirement on the environment in OE.InjDatSupp.

OE.Process-TOE requires the TOE Manufacturer to implement those measures assumed in P.Process-TOE. Therefore, the organisational security policy is covered by this objective, as far as organisational measures are concerned. The only issue not completely covered by these measures is the fact that the TOE has to support the possibility of unique identification. This is the content of O.Identification. Therefore, the organisational security policy is covered by OE.Process-Card and O.Identification. (Note that as in section 7.1 of [BSI-PP-0002] this applies to Phases 2-3 of the lifecycle.)

The basic rationale for T.Leak-Inherent, T.Phys-Probing, T.Phys-Manipulation, T.Malfunction, T.Leak-Forced, and T.Abuse-Func is given in section 7.1 of [BSI-PP-0002]: for all threats the corresponding objectives O.Leak-Inherent, O.Phys-Probing, O.Phys-Manipulation, O.Malfunction, O.Leak-Forced, and O.Abuse-Func are stated in a way, which directly corresponds to the description of the threat. It is clear from the description of each objective, that the corresponding threat is removed if the objective is valid. More specifically, in every case the ability to use the attack method successfully is countered, if the objective holds.

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. The text below gives further rationale from [PA].

Compared to [BSI-PP-0002] a clarification has been made for the security objective "Usage of Hardware Platform (OE.Plat-Appl)": If required the Smartcard Embedded Software shall use these cryptographic services of the TOE and their interface as specified. In addition, the Smartcard Embedded Software must implement functions which perform operations on keys (if any) in such a manner that they do not disclose information about confidential data. This addition ensures that the assumption A.Plat-Appl is still covered by the objective OE.Plat-Appl although additional functions are being supported according to O.Add-Functions.

Compared to [BSI-PP-0002] a clarification has been made for the security objective "Treatment of User Data (OE.Resp-Appl)": By definition, encryption data, plain text data, and cryptographic keys are User Data. So, the Smartcard Embedded Software will protect such data if required, and the keys and functions are used 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. These measures make sure that the assumption A.Resp-Appl is still covered by the security objective OE.Resp-Appl although additional functions are being supported according to P.Add-Functions.

The justification of the additional policy and the additional assumption show that they do not contradict to the rationale already given in [BSI-PP-0002] for the assumptions, policy and threats defined there.

Smartcard Embedded Software may implement measures using the ability given by the TOE to embedded software to detect and respond to the results of attacks based on these threats, in O.SWDetect and O.SWResponse. This can help address some of the core threats – T.Phys-Manipulation, T.Malfunction and T.Abuse-Func by detecting the results of attempts to tamper with the operation of the IC, and using additional defensive measures at the level of the target of the attack[10]. However, since no assumptions are made about the content of Smartcard Embedded

---

[10] An attacker only stands to gain in a material sense if the applications themselves are attacked, since these represent the only assets that yield direct benefits to the attacker.

Software (and hence the use made of these features), these objectives are not included for the core threats in the table above.

T.NoSWDetect and T.NoSWResponse are directly addressed by O.SWDetect and O.SWResponse respectively.

T.RND is addressed by O.RND.

8.2   Security Requirements Rationale

The way in which [BSI-PP-0002] objectives are implemented by SFRs and requirements on the environment is given in section 7.2 of [BSI-PP-0002]. The table below includes the mapping from section 7.2 of [BSI-PP-0002] and adds the rationale for the additional SFRs in this Security Target.

Table 8-2: Coverage of Objectives by SFRs

Objective	Addressed by SFR	Security Requirements for the Environment
O.Leak-Inherent	FDP_ITT.1 [HW], FDP_ITT.1 [ACL], FPT_ITT.1 [HW], FPT_ITT.1 [ACL], FDP_IFC.1 [HW], FDP_IFC.1 [ACL]	
O.Phys-Probing	FPT_PHP.3	
O.Phys-Manipulation	FPT_PHP.3	
O.Malfunction	FRU_FLT.2, FPT_FLS.1, FPT_SEP.1	
O.Leak-Forced	FDP_ITT.1 [HW], FDP_ITT.1 [ACL], FPT_ITT.1 [HW], FPT_ITT.1 [ACL], FDP_IFC.1 [HW], FDP_IFC.1 [ACL], FRU_FLT.2, FPT_FLS.1, FPT_SEP.1, FPT_PHP.3	
O.Abuse-Func	FMT_LIM.1, FMT_LIM.2, FDP_ITT.1 [HW], FDP_ITT.1 [ACL], FPT_ITT.1 [HW], FPT_ITT.1 [ACL], FDP_IFC.1 [HW], FDP_IFC.1 [ACL], FPT_PHP.3, FRU_FLT.2, FPT_FLS.1, FPT_SEP.1	

Objective	Addressed by SFR	Security Requirements for the Environment
O.Identification	FAU_SAS.1	
O.RND	FCS_RND.1 [TRNG], FCS_RND.1 [PRNG], FDP_ITT.1 [HW], FDP_ITT.1 [ACL], FPT_ITT.1 [HW], FPT_ITT.1 [ACL], FDP_IFC.1 [HW], FDP_IFC.1 [ACL], FPT_PHP.3, FRU_FLT.2, FPT_FLS.1, FPT_SEP.1	
O.Add-Functions	FDP_ITT.1 [ACL], FPT_ITT.1 [ACL], FDP_IFC.1 [ACL], FCS_COP.1 [RSA], FCS_COP.1 [3DES], FCS_COP.1 [SHA-1], FCS_COP.1 [SHA-256], FCS_COP.1 [RIPEMD-160], 4FCS_CKM.1 [RSA]	
O.SWDetect	FPT_FLS.1, FDP_ACC.1 [CRP], FDP_ACC.1 [WPP], FDP_ACF.1 [CRP], FDP_ACF.1 [WPP]	
O.SWResponse	FPT_FLS.1	
OE.Plat-Appl		RE.Phase-1
OE.Process-TOE		Assurance Components: refer to below ♣
OE.Process-Card		RE.Process-Card, possibly supported by RE.Phase-1
OE.Resp-Appl	FDP_ITC.1, FCS_CKM.1, FCS_CKM.4, FMT_MSA.2	RE.Phase-1
OE.InjDatSupp		RE.Phase-1

♣ Assurance Components: Delivery (ADO_DEL); Installation, generation, and startup (ADO_IGS) (using Administrator Guidance (AGD_ADM), User guidance (AGD_USR)); CM automation (ACM_AUT); CM Capabilities (ACM_CAP); CM Scope ACM_SCP); Development Security (ALC_DVS); Life Cycle Definition (ALC_LCD); Tools and Techniques (ALC_TAT)

Reference is made to section 7.2 of [BSI-PP-0002] for the basic rationale. The remainder of this section deals with the additional parts of the rationale introduced for this Security Target.

It is noted that the features covered by FDP_ACC.1 and FDP_ACF.1 address potential physical manipulation and malfunction attacks because they constrain the ways in which execution can occur. Furthermore, these access control measures directly protect parameters controlling some of the other security measures provided under FPT_FLS.1 from external attacks (e.g. only

Smartcard Embedded Software in ROM can set firewall management unit parameters, hence attacks based on changing these parameters in other conditions induced by attackers will be prevented).

O.Phys-Manipulation and O.Malfunction can be further addressed by Smartcard Embedded Software by using the TOE's features that allow embedded software to detect and respond to execution states that could represent attacks (included under FPT_FLS.1). However, this depends on the Smartcard Embedded Software and is therefore beyond the scope of the TOE.

The requirements of O.Add-Functions to provide scrambled copy, compare and XOR; to provide secure multiply; and to provide secure modular inversion, are covered by SFRs: FDP_ITT.1 [ACL], FPT_ITT.1 [ACL] and FDP_IFC.1 [ACL]; as these operations include transfer between separated parts of the TOE (between memory and operation units).

The cryptographic functions requirement of O.Add-Functions is directly implemented by FCS_COP.1.1. Additional security requirements on the environment, relating to the use of this functionality, are included in RE.Phase-1 and RE.Cipher. Rationale for this is excerpted from [PA] as follows.

The security functional requirement "Cryptographic operation (FCS_COP.1)" exactly requires those functions to be implemented which are demanded by O.Add-Functions. Therefore, FCS_COP.1 is suitable to meet the security objective.

Nevertheless, the developer of the Smartcard Embedded Software must ensure that the additional functions are used as specified and that the User Data processed by these functions are protected as defined for the application context. These issues are addressed by the requirement RE.Phase-1 and more specifically by the security functional requirements specified below.

- [FDP_ITC.1 Import of user data without security attributes or FCS_CKM.1 Cryptographic key generation],

- FCS_CKM.4 Cryptographic key destruction,

- FMT_MSA.2 Secure security attributes.

which will be fulfilled in the environment (addressed by the security environment objective OE.Resp-Appl), and the details are not known, with the following exceptions:

- The TOE itself fulfils FCS_CKM.1 [RSA] to meet the dependency of the RSA iteration of FCS_COP.1;

- The hash functions SHA-1, SHA-256 and RIPEMD-160 do not use cryptographic keys so the FCS_CKM.1 and FCS_CKM.4 dependencies for the SHA-1, SHA-256 and RIPEMD-160 iterations of FCS_COP.1 can be considered to be automatically discharged.

The security functional requirements required to meet 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. However, key-dependent functions could be implemented in the Smartcard Embedded Software. In this case RE.Cipher requires that these functions ensure that confidential data (User Data) cannot be disclosed while they are just being processed by the Smartcard Embedded Software. Therefore, with respect to the Smartcard Embedded Software the issues addressed by the objectives just mentioned are addressed by the requirement RE.Cipher.

The use of cryptographic algorithms requires use of appropriate keys - otherwise they do not provide security. The requirement RE.Cipher addresses these specific issues since cryptographic keys and other data are provided by the Smartcard Embedded Software. RE.Cipher requires that keys must be kept confidential. They must be unique with a very high probability, cryptographically strong etc. If keys are imported into the TOE (usually after TOE Delivery), it must be ensured that quality and confidentiality is maintained. Therefore, with respect to the environment the issues addressed (i) by the objectives just mentioned and (ii) implicitly by O.Add-Functions, are addressed by the requirement RE.Cipher.

The justification of the security objective O.Add-Functions and the additional requirements (both for the TOE and its environment) show that they do not contradict to the rationale already given in [BSI-PP-0002] for the assumptions, policy and threats defined there.

O.SWDetect is implemented by the exception conditions recognised under FPT_FLS.1 and the opportunities provided to test for violations of secure state (or to maintain velocity check parameters) by the EEPROM write interrupts in FPT_FLS.1. As noted for O.Phys-Manipulation and O.Malfunction above, the access control provisions in FDP_ACC.1 and FDP_ACF.1 provide protection for the ways in which these detection features are used.

O.SWResponse is implemented by the ability of Smartcard Embedded Software to cause a reset (by setting the halt bit to halt the processor, as noted under FPT_FLS.1), and the opportunity for other protective measures as part of the EWE interrupt routines provided under FPT_FLS.1.

RE.Phase-1 contributes to OE.InjDatSupp. It is because the Smartcard Embedded Software is required by RE.Phase-1 to have a certain configuration so that the injected data shall be generated, distributed, maintained and destroyed in an adequately secure fashion.

The assignment/selection operations performed on the SFRs drawn from [BSI-PP-0002] are shown in [BSI-PP-0002] itself. The additional operations performed in this ST are as follows:

Table 8-3: Completion of SFRs

SFR	Operation required	Operation performed
FCS_RND.1 [TRNG]	[assignment: a defined quality metric]	The requirements of the monobit, poker, runs, long run, and autocorrelation tests in [AIS31].
FCS_RND.1 [PRNG]	[assignment: a defined quality metric]	The requirements of the statistical and formal tests to meet functionality class K3 as defined in [AIS20]
FCS_COP.1 [3DES]	[assignment: list of cryptographic operations]	Encryption and decryption
	[assignment: cryptographic algorithm]	3DES
	[assignment: cryptographic key sizes]	112 bit
	[assignment: list of standards]	U.S. Department of Commerce / National Bureau of Standards Data Encryption Standard (DES), FIPS PUB 46-3, 1999 October 25

SFR	Operation required	Operation performed
FCS_COP.1 [RSA]	[assignment: list of cryptographic operations]	Encryption and decryption
	[assignment: cryptographic algorithm]	RSA and RSA CRT
	[assignment: cryptographic key sizes]	1024-2048
	[assignment: list of standards]	ISO/IEC 9796-2, Annex A
FCS_COP.1 [SHA-1]	[assignment: list of cryptographic operations]	hashing
	[assignment: cryptographic algorithm]	SHA-1
	[assignment: cryptographic key sizes]	Not applicable to hashing
	[assignment: list of standards]	Secure Hash Standard, Federal Information Processing Standards Publication 180-2, 2002
FCS_COP.1 [SHA-256]	[assignment: list of cryptographic operations]	hashing
	[assignment: cryptographic algorithm]	SHA-256
	[assignment: cryptographic key sizes]	Not applicable to hashing
	[assignment: list of standards]	Secure Hash Standard, Federal Information Processing Standards Publication 180-2, 2002
FCS_COP.1 [RIPEMD-160]	[assignment: list of cryptographic operations]	Hashing
	[assignment: cryptographic algorithm]	RIPEMD-160
	[assignment: cryptographic key sizes]	Not applicable to hashing
	[assignment: list of standards]	ISO/IEC FDIS 10118-3, 2003: Information Technology – Security techniques – Hash-functions – Part 3
FCS_CKM.1 [RSA]	[assignment: cryptographic key generation algorithm]	RSA
	[assignment: cryptographic key sizes]	1024-2048
	[assignment: list of standards]	ISO/IEC 9796-2, Annex A
FDP_ACC.1 [CRP]	[assignment: access control SFP]	Controlled-Register Policy
	[assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP].	Any attempt to set bits in defined, security-relevant registers
FDP_ACC.1 [WPP]	[assignment: access control SFP]	Write-Protect Policy
	[assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP].	Software to write to EEPROM.

SFR	Operation required	Operation performed
FDP_ACF.1 [CRP]	[assignment: access control SFP]	Controlled-Register Policy
	[assignment: security attributes, named groups of security attributes]	The memory area where the software is executed from, the memory area to be accessed and the operation to be performed
	[assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]	Evaluate the corresponding permission control information: "The ability to set the values in the controlled registers is restricted to software operations executed from the ROM." before, during or after the access so that accesses to be denied can not be utilised by the subject attempting to perform the operation
	[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]	None
	[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]	None
FDP_ACF.1 [WPP]	[assignment: access control SFP]	Write-Protect Policy
	[assignment: security attributes, named groups of security attributes]	The memory area to be accessed and the operation to be performed
	[assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]	Evaluate the corresponding permission control information: "An attempt to write to EEPROM shall fail, leaving the previous memory contents unaltered if protection is set for the relevant EEPROM page. Once a page is write-protected, this protection cannot be removed." before, during or after the access so that accesses to be denied can not be utilised by the subject attempting to perform the operation
	[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]	None
	[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]	None

8.2.1   Dependencies

The basic dependencies are shown in section 7.2.2 of [BSI-PP-0002] and are applicable to this ST – these are summarised in the table below:

Table 8-4: Dependencies from [BSI-PP-0002, 7.2.2]

SFR	Dependencies	Fulfilled by Security Requirements in [BSI-PP-0002]?
FRU_FLT.2	FPT_FLS.1	Yes
FPT_FLS.1	ADV_SPM.1	Yes (Part of EAL4)
FPT_SEP.1	None	No dependency
FMT_LIM.1	FMT_LIM.2	Yes
FMT_LIM.2	FMT_LIM.1	Yes
FAU_SAS.1	None	No dependency
FPT_PHP.3	None	No dependency
FDP_ITT.1 [HW]	FDP_ACC.1 or FDP_IFC.1	Yes
FDP_IFC.1 [HW]	FDP_IFF.1	See discussion in [BSI-PP-0002, 7.2.2]
FPT_ITT.1 [HW]	None	No dependency
FDP_ITT.1 [ACL]	FDP_ACC.1 or FDP_IFC.1	Yes
FDP_IFC.1 [ACL]	FDP_IFF.1	See discussion in [BSI-PP-0002, 7.2.2]
FPT_ITT.1 [ACL]	None	No dependency
FCS_RND.1 [TRNG]	None	No dependency
FCS_RND.1 [PRNG]	None	No dependency

The additional dependencies relating to the new SFRs introduced in this ST are analysed below.

Table 8-5: Additional SFR Dependencies

SFR	Dependencies	Fulfilled by Security Requirements in this ST?
FCS_COP.1 [3DES]	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	Yes (by the environment)
FCS_COP.1 [RSA]	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	Yes (by FCS_CKM.1 [RSA], and for others by the environment)
FCS_COP.1 [SHA-1]	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	Yes (by the environment)
FCS_COP.1 [SHA-256]	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	Yes (by the environment)
FCS_COP.1 [RIPEMD-160]	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	Yes (by the environment)
FCS_CKM.1 [RSA]	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4 FMT_MSA.2	Yes (by FCS_COP.1 [RSA], and for others by the environment)
FDP_ITC.1	[FDP_ACC.1, or FDP_IFC.1] FMT_MSA.3	No additional requirement – see discussion below

SFR	Dependencies	Fulfilled by Security Requirements in this ST?
FCS_CKM.1	FCS_CKM.4 FMT_MSA.2	No additional requirement – see discussion below
FCS_CKM.4	[FDP_ITC.1 or FCS_CKM.1] FMT_MSA.2	No additional requirement – see discussion below
FMT_MSA.2	ADV_SPM.1 [FDP_ACC.1 or FDP_IFC.1] FMT_MSA.1 FMT_SMR.1	No additional requirement – see discussion below
FDP_ACC.1 [CRP]	FDP_ACF.1	Yes
FDP_ACC.1 [WPP]	FDP_ACF.1	Yes
FDP_ACF.1 [CRP]	FDP_ACC.1	Yes
	FMT_MSA.3	No additional requirement – see discussion below
FDP_ACF.1 [WPP]	FDP_ACC.1	Yes
	FMT_MSA.3	No additional requirement – see discussion below

The dependencies defined for FCS_COP.1 in [CC/2] are discharged by FCS_CKM.1 [RSA] implemented in the TOE, and the requirements on the environment, as described in [PA].

Hence there is no further functional requirement on the TOE arising from the dependencies of FCS_COP.1.

The dependencies defined for FDP_ITC.1, FCS_CKM.1, FCS_CKM.4, and FMT_MSA.2 are not resolved because they will be fulfilled in the environment, where the appropriate decisions will be made.

The dependency defined for FDP_ACF.1 in [CC/2] is discharged as follows:

- FDP_ACF.1 [CRP] is simply applied for the control of the access to registers, and it does not require setting of the initial value of security attributes, which is required under FMT_MSA.3. Also, FDP_ACF.1 [WPP] simply covers the control of the access to the EEPROM, and it also does not require setting of the initial value of security attributes, which is required under FMT_MSA.3. Therefore, the dependency on the component FMT_MSA.3 is discharged.

The discussion in sections 8.2 and 8.3, and the rationale in section 7.2 of [BSI-PP-0002], show how the security functional requirements support each other in meeting the security objectives of this ST. Together with the discussion of dependencies above this shows that the security functional requirements build a mutually supportive whole.

8.3 TOE Summary Specification Rationale

The table below shows the ways in which the SFRs are implemented by TOE security functions.

Table 8-6: SFR Mapping to TOE Security Functions

SFR	TOE Security Function
FRU_FLT.2	SF.HWProtect, SF.FMU, SF.ESFunctions
FPT_FLS.1	SF.HWProtect, SF.FMU, SF.ESFunctions, SF.EEPAccess
FPT_SEP.1	SF.HWProtect, SF.FMU, SF.ESFunctions, SF.EEPAccess
FMT_LIM.1	SF.TestModeControl
FMT_LIM.2	SF.TestModeControl
FAU_SAS.1	SF.Inject
FPT_PHP.3	SF.HWProtect, SF.FMU, SF.ESFunctions
FDP_ITT.1 [HW]	SF.LeakProtect
FPT_ITT.1 [HW]	SF.LeakProtect
FDP_IFC.1 [HW]	SF.LeakProtect
FDP_ITT.1 [ACL]	SF.ACL-LeakProtect
FPT_ITT.1 [ACL]	SF.ACL-LeakProtect
FDP_IFC.1 [ACL]	SF.ACL-LeakProtect
FCS_RND.1 [TRNG]	SF.RNG
FCS_RND.1 [PRNG]	SF.RNG
FCS_COP.1 [3DES]	SF.DES
FCS_COP.1 [RSA]	SF.MMCopro
FCS_COP.1 [SHA-1]	SF.Hash
FCS_COP.1 [SHA-256]	SF.Hash
FCS_COP.1 [RIPEMD-160]	SF.Hash
FCS_CKM.1 [RSA]	SF.MMCopro
FDP_ACC.1 [CRP]	SF.FMU
FDP_ACC.1 [WPP]	SF.EEPAccess
FDP_ACF.1 [CRP]	SF.FMU
FDP_ACF.1 [WPP]	SF.EEPAccess

Details of the TOE summary specification rationale are not given in this version of the Security Target.

8.4 Assurance Requirements and Strength of Function Rationale

This ST follows the rationale given in section 7.2.3 of [BSI-PP-0002] for the choice of EAL4, assurance augmentations and the strength of function SOF-high.

8.5 Mutual Support and Internal Consistency

The discussion of security functional requirements, TOE Summary Specification and assurance components in the sections above shows that mutual support and consistency are present for both groups of requirements. The arguments given for the adequacy of the assurance components for

the TOE functionality also shows that the functional and assurance requirements support each other and that there are no inconsistencies between these groups.

For the additional functionality included in O.Add-Functions, the security functional requirements required to meet the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced also protect the cryptographic algorithms implemented according to the security functional requirement FCS_COP.1. Therefore, these security functional requirements support the secure implementation and operation of FCS_COP.1.

8.6   PP Claims Rationale

This ST implements all of the requirements of [BSI-PP-0002] by inclusion (as shown in each of the relevant sections), and hence no further rationale is required.

** End of Document **