

**RS45C Version 01**  
**Security Target**  
**-Public Version-**

**Renesas Electronics Corporation**

Revision 1.0

4 June, 2010



## 0. History

### 0.1 Revision

Revision	Date	Section	Description
Rev. 1.0	4 June, 2010	-	Release

### 0.3 Abbreviations

Term	Meaning
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 security IC.
CPU	Central Processing Unit
CRAM	RAM dedicated for use by MMC coprocessor
CRC	Cyclic Redundancy Check
DES	Data Encryption Standard
DFA	Differential Fault Analysis
EAL	Evaluation Assurance Level
ECB	Electronic Code Book A mode of DES and 3DES encryption.
EEPROM	Electrically Erasable Programmable Read-Only Memory
HW RNG	Hardware random number generator (physical random number generator)
IC	Integrated Circuit
ISC	Integrated Security Concept
IT	Information Technology
MCU	Micro Computer Unit
MMC	Modular Multiplication Coprocessor
PKI	Public Key Infrastructure
PP	Protection Profile
PRNG	Pseudo Random Number Generator
RAM	Random Access Memory
RL	Random Logic (Glue Logic)
RNG	Random Number Generator
ROM	Read-Only Memory
RSA	Rivest, Shamir, Adelman – a public key encryption algorithm, named after its inventors.
SFP	Security Function Policy
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
WAP	Wireless Application Protocol

## 0.4 Glossary

Term	Meaning
Embedded Software	<p>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:</p> <ul style="list-style-type: none"> <li>• Hard-coded – held in ROM</li> <li>• Soft-coded – held in EEPROM or RAM.</li> </ul> <p>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.</p> <p>Note that Embedded Software includes all software on a TOE other than the IC Dedicated Software.</p> <p>Embedded Software is also referred to as ‘Security IC Software’ (especially in [BSI-PP-0035]).</p>
EWE	An interrupt generated by the TOE whenever an attempt is made to write to EEPROM.
IC Dedicated Software	Software developed by Renesas and embedded in the IC. (Adopted from [BSI-PP-0035])
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 1.4.3).
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 Security IC Embedded Software present, and the versions of masks and specifications applicable.
Option List	<p>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 :</p> <ul style="list-style-type: none"> <li>• Selection of whether pre-personalisation data injection is required</li> </ul> <p>The option list also describes the content and structure of the manufacturing identification data that Renesas will inject (see section 1.4.4.2).</p>
Renesas	Refer to Renesas Electronics Corporation. ( <a href="http://www.renesas.com/">http://www.renesas.com/</a> )
Reset state	<p>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.</p> <p>See also section 7.1.</p>
Smartcard	Composition of the TOE, the Smartcard Embedded Software, User Data and the package (the smartcard carrier).
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 1.4.3. 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).
TOE Manufacturer	<p>(As defined in section 8.7 of [BSI-PP-0035]) 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.</p> <p>For the RS45C, the TOE Manufacturer refers to Renesas.</p>
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 Security IC Embedded Software in the application context. User data comprises all data in the final Security 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.
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 .....	1
1.1	ST Reference .....	1
1.2	TOE Reference .....	1
1.3	TOE Overview .....	2
1.4	TOE Description .....	3
1.4.1	RS45C Product Description .....	3
1.4.2	TOE Intended Usage .....	7
1.4.3	TOE Lifecycle .....	7
1.4.4	TOE Environments .....	10
2.	Conformance Claims .....	12
2.1	CC Conformance Claim .....	12
2.2	PP Claims .....	12
2.2.1	PP Reference .....	12
2.2.2	PP Tailoring .....	12
2.2.3	PP Additions .....	12
2.3	Package Claim .....	12
2.4	Conformance Rationale .....	12
2.4.1	CC Conformance Rationale .....	12
2.4.2	PP Claim Rationale .....	13
2.4.3	Package Claims Rationale .....	13
3.	Security Problem Definition .....	14
3.1	Description of Assets .....	14
3.2	Threats .....	15
3.2.1	Threats Defined in [BSI-PP-0035] .....	15
3.2.2	Other Threats .....	18
3.3	Organisational Security Policies .....	18
3.3.1	Policy Requirement from [BSI-PP-0035] .....	18
3.3.2	Policy Requirement from [PA] .....	19
3.4	Assumptions .....	19
3.4.1	Assumptions from [BSI-PP-0035] .....	19
3.4.2	Assumptions from [PA] .....	21
3.4.3	Other Assumptions .....	21
4.	Security Objectives .....	22
4.1	Security Objectives for the TOE .....	22
4.1.1	Objectives from [BSI-PP-0035] .....	22
4.1.2	Objectives Based on [PA] .....	25
4.1.3	Other Objectives .....	25
4.2	Security Objectives for the Environment .....	25
4.2.1	Security objectives for the security IC Embedded software development environment from [BSI-PP-0035] .....	25
4.2.2	Security Objectives for the Operational Environment from [BSI-PP-0035] .....	27
4.2.3	Other Environment Security Objectives .....	27
4.3	Security Objectives Rationale .....	27
5.	Extended Components Definition .....	30
5.1	Extended Components Definition from [BSI-PP-0035] .....	30
5.1.1	Definition of the Family FCS_RNG .....	30
5.1.2	Definition of the Family FMT_LIM .....	30

5.1.3	Definition of the Family FAU_SAS.....	30
6.	Security Requirements .....	31
6.1	Security Functional Requirements .....	31
6.1.1	Security Functional Requirements from [BSI-PP-0035] .....	31
6.1.2	Security Functional Requirements Based on [PA].....	36
6.2	Security Assurance Requirements.....	37
6.2.1	Refinements of the TOE Security Assurance Requirements.....	38
6.2.2	Refinements regarding CM scope (ALC_CMS).....	38
6.2.3	Functional specification (ADV_FSP) .....	38
6.2.4	Rationale for the Assurance Requirements .....	39
6.3	Security Requirement Rationale .....	40
6.3.1	Rational for the Security Functional Requirements .....	40
6.3.2	Dependencies of Security Functional Requirements .....	42
7.	TOE Summary Specification .....	45
7.1	TOE Security Functionalities.....	45
7.2	Correspondence between TOE Security Functionalities and SFR.....	47
7.3	TOE Summary Specification Rationale .....	48
8.	Reference .....	49
8.1	Reference Materials .....	49
8.2	Others .....	49

### List of Figures

Figure 1-1:	Configuration of the TOE .....	2
Figure 1-2:	Internal Block Diagram of the TOE .....	4
Figure 1-3:	Design and Manufacturing Lifecycle .....	8
Figure 6-1:	Paradigm Regarding Operating Conditions.....	32

### List of Tables

Table 1-1:	TOE Configuration.....	1
Table 4-1:	Coverage of Security Assumptions, Policies and Threats by Objectives .....	28
Table 6-1:	Assurance Components.....	37
Table 6-2:	Security Assurance Requirements, overview of differences of refinements .....	38
Table 6-3:	Security Requirements versus Security Objectives.....	40
Table 6-4:	Completion of SFRs.....	42
Table 6-5:	Dependencies of Security Functional Requirements .....	43
Table 6-6:	Additional SFR Dependencies .....	43
Table 7-1:	TOE Security Functionalities Mapping to SFRs.....	47
Table 7-2:	SFR Mapping to TOE Security Functionalities .....	48

# 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 Reference

Title: RS45C Version 01 Security Target -Public version-

Revision: 1.0

Provided by: Renesas Electronics Corporation.

This Security Target applies to the Renesas RS45C integrated circuit (version 01) (as defined in detail in the configuration list for the evaluation).

*This public version of the RS45C Security Target (also known as “ST-lite”) is abridged from the evaluated version of the full Security Target (version 3287) with the approval of the Certification Body.*

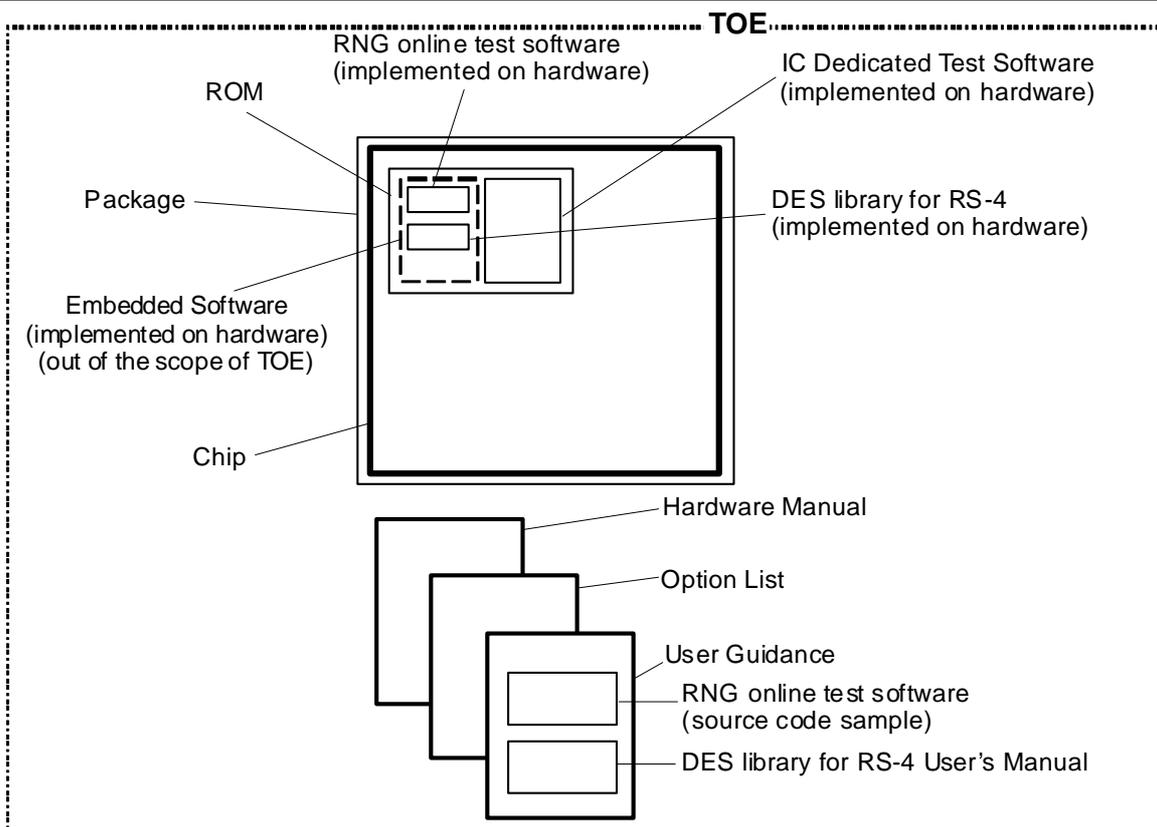
## 1.2 TOE Reference

The TOE configuration is summarised in the table below:

**Table 1-1: TOE Configuration**

Item Type	Item	Version	Form of delivery
Hardware	RS45C integrated circuit	01	Wafer or packaged module (see section 1.4.3)
Software	IC Dedicated Test Software Test ROM software	2080	Included in RS45C test ROM
Software	RNG on-line test software	1.30 (defined by the version of [UGM])	Hardcopy: provided as a part of [UGM]. (This is implemented in the Embedded Software by the user)
Software	DES library for RS-4		Electronic data/ Hardcopy: provided as a part of [UGM]. (This is implemented in the Embedded Software by the user)
	RS4_LL.lib	3268	
	RS4_LL.txt	3268	
	RS4_LL.h	3292	
Document	Hardware Manual	1.00	Electronic data/Hardcopy
Document	User Guidance	1.30	Electronic data/Hardcopy
Document	Option List	1.1	Electronic data/Hardcopy

Further description of the TOE is provided in section 1.4.1.



**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. Among software, the Embedded Software is outside the scope of the TOE. The RNG online test software is provided to the developer of the Embedded Software as a source code sample in [UGM], and it is implemented in the TOE as the embedded software by the Embedded Software developer.

### 1.3 TOE Overview

This document is the ST for the Renesas RS45C IC product, intended for use as a Security IC. The TOE complies with the Eurosmart Protection Profile developed by the Secure Semiconductor Vendor Group [BSI-PP-0035]. 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 Security IC 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 Composite Product 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 coprocessor and power analysis attack protection are all included providing a strong on-chip hardware security structure.

---

Uniquely, Renesas Security ICs 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 security 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.

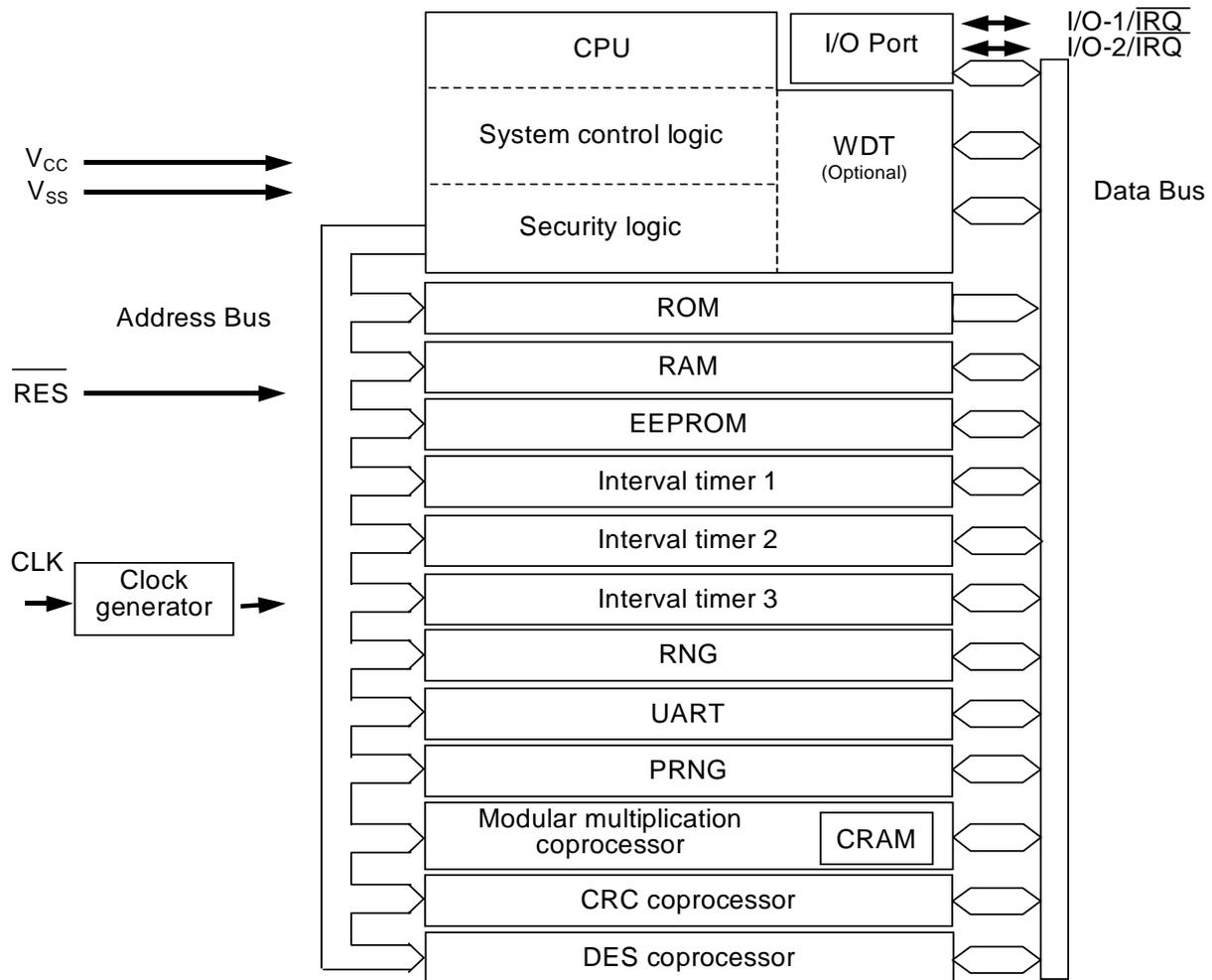
## **1.4 TOE Description**

### **1.4.1 RS45C Product Description**

The TOE consists of the hardware shown in Figure 1-2, 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 3.2 of [BSI-PP-0035].

A block diagram of the chip is shown in Figure 1-2 below:



**Figure 1-2: Internal Block Diagram of the TOE**

#### 1.4.1.1 Hardware

The TOE is an integrated circuit based around the high-speed RS-4 Series CPU core (derived from Renesas' widely used H8S general purpose core). As can be seen from the block diagram in Figure 1-2, the MCU comprises the following major blocks in addition to the CPU: ROM, RAM, EEPROM, random number generator (RNG), MMC, DES coprocessor, CRC coprocessor, UART, three interval timers, WDT (optional), 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 and is upwards compatible with the H8/300H core

- The instruction set is implemented with 16-bit variable instruction lengths (2 to 10 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, or 16 x 16-bits)
- Signed or unsigned divide instruction (16 ÷ 8, 16 ÷ 16, 32 ÷ 16-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: 224kbytes
- RAM: 6kbytes + 2 kbytes of coprocessor RAM
- EEPROM: 36kbytes
  - on-chip charge pump and independent oscillator
  - special write instruction, and interrupt generation on writing
  - page write/erase
  - individual page protection

**RNG:** this can generate 16-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. 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 – 2624 bits. *As no cryptographic software or algorithm is provided by Renesas for the TOE, the functionality of this coprocessor is included in the evaluated hardware configuration, but its specific use in cryptographic software is not included.*

**DES coprocessor:** this hardware engine can be used to provide either DES or triple-DES functions. 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 Security IC Embedded Software. Only triple-DES is claimed as security functionality for this product. For some application contexts, Single-DES may be sufficient<sup>1</sup> - this is a matter for the Security IC Embedded Software security target.

**CRC coprocessor:** CRC (Cyclic Redundancy Check) coprocessor generates codes for detecting errors in data blocks. The CRC codes are created using a generator polynomial  $CRC-CCITT(X^{16}+X^{12}+X^5+X^0)$ .

**Interval timer:** the TOE has interval timer 1, interval timer 2, and interval timer 3. 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 in [OPT].

---

<sup>1</sup> Although strength of cryptographic functions is beyond the scope of a Common Criteria evaluation, triple DES would probably be required to be resistant to attacks performed by an attacker possessing High attack potential. Therefore only triple-DES is claimed as a security function.

---

**I/O:** I/O comprises I/O-1/IRQ and I/O-2/IRQ (see Figure 1-2). 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. Full-duplex asynchronous mode that conforms to the ISO/IEC CD standard 10536-4:

**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

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].

#### 1.4.1.2 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.

##### **The RNG On-line Test Software:**

The RNG On-line Test Software is provided to perform the on-line test for randomness, as required in [AIS31]. To enable users to deploy the software as necessary, this software is supplied as a listing in [UGM]. Note that the use of this software is part of the intended method of use of the TOE under certain conditions, and it is therefore part of the evaluated configuration.

##### **DES Library for RS-4:**

In order to make compliance with the device's security countermeasure requirements as straightforward as possible for the user, DES Library for RS-4 is provided. Guidelines for its use are given in [UGM].

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

---

### **1.4.1.3 Documents**

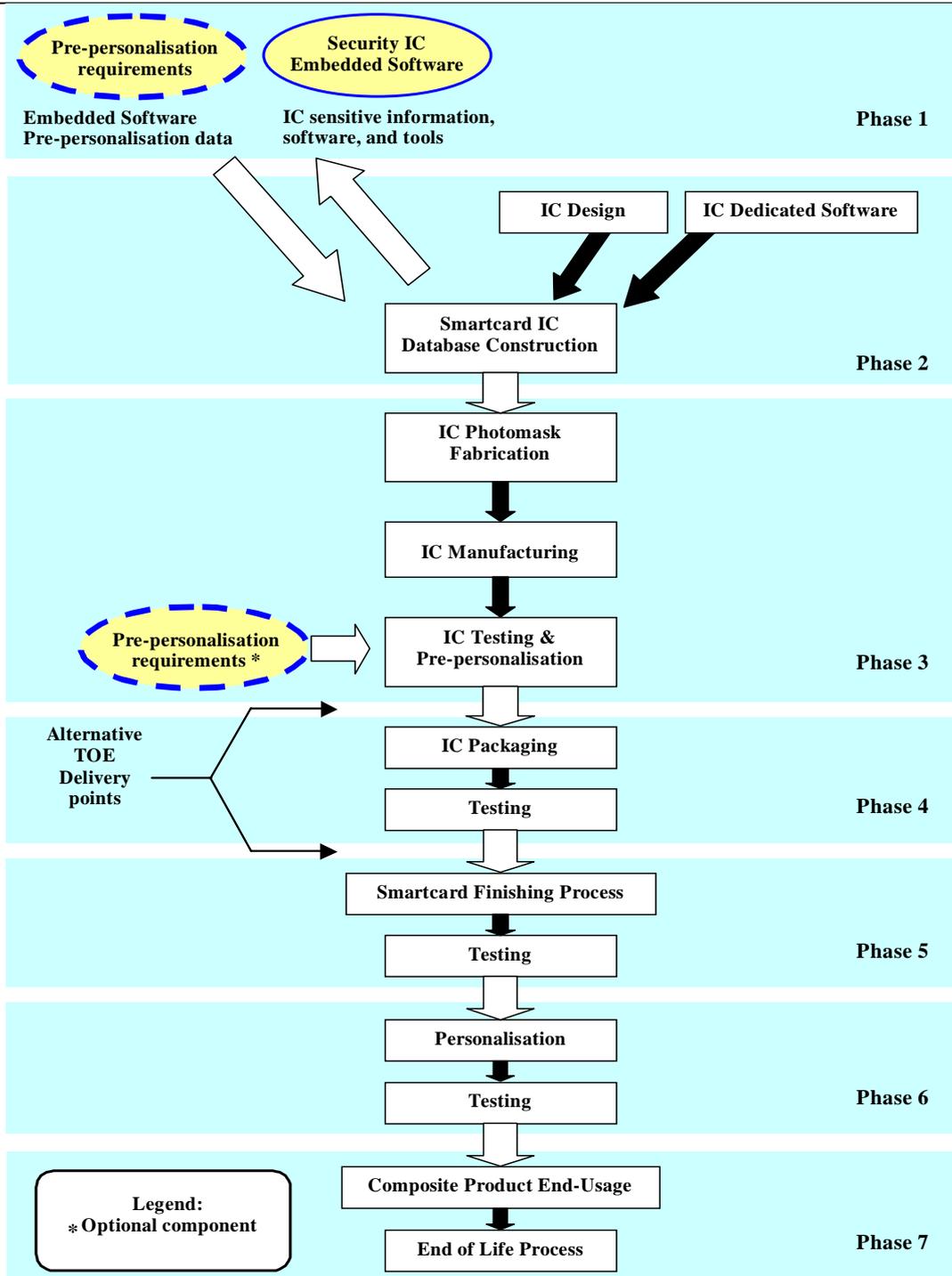
The TOE Hardware Manual [HM] is supplied as the basic reference for users who are developing Security IC Embedded Software. Guidance for the secure use of RS45C in applications is given in the User Guidance Manual [UGM]. Options and contents of the identification data stored in the EEPROM are described in [OPT].

### **1.4.2 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.

### **1.4.3 TOE Lifecycle**

The design and manufacturing lifecycle for the TOE is shown in Figure 1-3 (and in section 7.1.1 of [BSI-PP-0035]). 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 1-3: Design and Manufacturing Lifecycle**

The stages shown are as listed below. This Security Target addresses phases 2 and 3 in Figure 1-3. 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: Security IC 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 Security IC 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 via a secure route to Renesas for injection during the manufacturing stage. Injection of data is described further in section 1.4.4.2. Secure receipt and handling of this data by Renesas is included within the scope of this security target.

- Phase 2: IC Development – this includes system design through logic, circuit and layout design. The Dedicated Test Software is also developed in this phase, to enable testing of the IC at various phases of its manufacture. The Security IC Embedded Software is received from 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: IC Manufacturing – The masks used to manufacture the various IC layers are created from the layout design during phase 2; ROM masks for the Security IC 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 Security IC 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 1.4.4.2).

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.
- Phases 5-7: Composite Product Integration, Personalisation and Operational-usage – these stages, and their security features, are determined by the customer, and are outside the scope of this security target.

### 1.4.3.1 Test and User Modes

During manufacturing and production (phases 2-4), the chip makes a series of 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 Security IC 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 from User Mode to Test Mode is designed to be very difficult.

### **1.4.3.2 TOE Delivery**

As noted above, the TOE is 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.

## **1.4.4 TOE Environments**

### **1.4.4.1 Development Environment**

Renesas' development environment for the TOE has implemented security measures specifically to ensure the security of the TOE and of Security IC Embedded Software and injection data used in manufacturing ICs for customers. As indicated in section 1.4.3, 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 customer ROM code and injection data (i.e. the Security IC 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 Security IC 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 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.

Security IC 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 Security IC design information. As a further measure, the group handling customer software is separate from the IC design team.

#### **1.4.4.2 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 Composite Product embedded with Security IC 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].

## **2. Conformance Claims**

### **2.1 CC Conformance Claim**

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

Because the ST conforms to [BSI-PP-0035], it includes extended functionality classes defined in section 5 of [BSI-PP-0035]. The ST is therefore [BSI-PP-0035] 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 EAL5 augmented (for augmentations see section 6.2).

### **2.2 PP Claims**

#### **2.2.1 PP Reference**

This ST conforms to [BSI-PP-0035].

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

#### **2.2.2 PP Tailoring**

FCS\_RNG.1 is completed with a quality metric – see section 6.1.1.5.

FAU\_SAS.1 is completed with a quality metric – see section 6.1.1.2.

#### **2.2.3 PP Additions**

The inclusions from [BSI-PP-0035] are clearly shown in the relevant section titles. All other threats, assumptions, objectives, extended components, and SFRs, in sections 3.2.2, 3.3.2, 3.4.2, 3.4.3, 4.1.2, 4.1.3, 4.2.3, and 6.1.2 are additional to those in the PP.

### **2.3 Package Claim**

The assurance level for this Security Target is EAL5 augmented. The augmentations to EAL5 are ALC\_DVS.2, and AVA\_VAN.5.

### **2.4 Conformance Rationale**

#### **2.4.1 CC Conformance Rationale**

This ST implements all of the requirements of [CC/1], [CC/2] and [CC/3] by inclusion (as shown in each of the relevant sections), and hence no further rationale is required.

---

## **2.4.2 PP Claim Rationale**

This ST for the TOE type as described in section 1.3 implements all of the requirements, security problem definition, objectives and security requirements, of [BSI-PP-0035] by inclusion (as shown in each of the relevant sections), and hence no further rationale is required.

## **2.4.3 Package Claims Rationale**

This ST implements all of the requirements of EAL5 augmented.

[BSI-PP-0035] requires the assurance level EAL4 augmented. Regarding the Application Note 21 of [BSI-PP-0035] the changes which are needed for EAL5 are described in the different relevant sections of this Security Target.

### 3. Security Problem Definition

#### 3.1 Description of Assets

This section defines the assets to be protected by the TOE. Section 3.1 of [BSI-PP-0035] gives the assets relating to the threats, and these are summarised below.

The assets to be protected are:

- the User Data  
this includes injection/pre-personalisation data and data generated and managed by the Security IC Embedded Software (subject to adequate protection by the software, see A.Key-Function, A.Plat-Appl and A.Resp-Appl in section 3.4)
- the Security IC Embedded Software stored and in operation, 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 asset is:

- the security services provided by the TOE for the Security IC Embedded Software

In particular integrity of the Security IC Embedded Software means that the Security IC 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 asset is:

- the random numbers generated by the TOE<sup>2</sup>

To be able to protect these assets the TOE shall protect its security functionality. Therefore critical information about the TOE shall be protected. Critical information includes:

- logical design data, physical design data, IC Dedicated Software, and configuration 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.

---

<sup>2</sup> 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.

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

## 3.2 Threats

### 3.2.1 Threats Defined in [BSI-PP-0035]

This section adopts the threats to ICs defined in section 3.2 of [BSI-PP-0035].

The TOE has the following high-level security concerns, as in section 3.1 of [BSI-PP-0035]:

- SC1 integrity of User Data and of the Security IC Embedded Software (while being executed/processed and while being stored in the TOE's memories).
- SC2 disclosure of User Data and of the Security IC Embedded Software (while being processed and while being stored in the TOE's memories).
- SC3 correct operation of the security services provided by the TOE for the Security IC Embedded Software.
- SC4 deficiency of random numbers.

The above 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 Security IC Embedded Software and is not a success for the attacker in itself.

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

- Phase 1 and the 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 up to TOE Delivery are covered by an organisational security policy

#### 3.2.1.1 Standard Threats

See section 3.2 of [BSI-PP-0035], and the example attack scenarios in section 7.3 of [BSI-PP-0035]. 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 Security IC in order to disclose confidential User Data as part of the assets.

No direct contact with the Security IC internals is required here. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements.

**T.Phys-Probing**

## Physical Probing

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

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

Physical probing requires direct interaction with the Security IC internals. Techniques commonly employed in IC failure analysis and IC reverse engineering efforts may be used. Before that, hardware security mechanisms and layout characteristics need to be identified. Determination of software design, including treatment of User Data may also be a pre-requisite.

**T.Malfunction**

## Malfunction due to Environmental Stress

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

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

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

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

**T.Phys-Manipulation**

## Physical Manipulation

An attacker may physically modify the Security IC in order to

- (i) modify User Data,
- (ii) modify the Security IC Embedded Software
- (iii) modify or deactivate security services of the TOE, or
- (iv) modify security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software.

The modification may be achieved through techniques commonly employed in IC failure analysis and IC reverse-engineering. The modification may result in the deactivation of a security feature. 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 (refer to T.Malfunction), the attacker requires to gather significant knowledge about the TOE's internal construction here.

**T.Leak-Forced**                      Forced Information Leakage

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

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

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

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 1.4.3).

### 3.2.1.2                      Threats Related to Security Services

**T.RND**                              Deficiency of Random Numbers

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

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

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

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.

### 3.2.2 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:

SC5 attacks on the Security IC Embedded Software may be made, and the software may not be able to respond to the attacks.

**T.NoSWResponse** Inability of Security IC Embedded Software to respond to an attack

If Security IC 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 Security IC 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).

## 3.3 Organisational Security Policies

### 3.3.1 Policy Requirement from [BSI-PP-0035]

The following policy requirement is taken from section 3.3 of [BSI-PP-0035].

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

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-0035].

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 1.4.4.2. 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 Security IC Embedded Software contains functionality to authenticate the IC<sup>3</sup>.

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.

### 3.3.2 Policy Requirement from [PA]

As an additional policy, the TOE provides specific security functionality which can be used by the Security IC 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 because it can only be decided in the context of the Composite Product application, against which threats the Security IC 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 Security IC Embedded Software:

- Triple Data Encryption Standard (3DES)

## 3.4 Assumptions

### 3.4.1 Assumptions from [BSI-PP-0035]

Appropriate "Protection during Packaging, Finishing and Personalisation (A.Process-Sec-IC)" must be ensured after TOE Delivery up to the end of Phase 6, as well as during the delivery to Phase 7 as specified below.

**A.Process-Sec-IC** Protection during Packaging, Finishing and Personalisation

It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use).

---

<sup>3</sup> For example, a hash or digital signature over a known area of memory might be provided by software.

---

This means that the Phases after TOE Delivery are assumed to be protected appropriately.

The information and material produced and/or processed by the Security IC Embedded Software Developer in Phase 1 and by the Composite Product Manufacturer can be grouped as follows:

- the Security IC Embedded Software including specifications, implementation and related documentation,
- pre-personalisation and personalisation data including specifications of formats and memory areas, test related data,
- the User Data and related documentation, and
- material for software development support

as long as they are not under the control of the TOE Manufacturer.

The developer of the Security IC Embedded Software must ensure the appropriate “Usage of Hardware Platform (A.Plat-Appl)” while developing this software in Phase 1 as specified below.

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

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

- (i) TOE guidance documents (refer to the Common Criteria assurance class AGD) such as the hardware data sheet, and the hardware application notes:  
i.e. TOE hardware manual [HM], user guidance manual [UGM], and the hardware application notes
- (ii) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as documented in the certification report.

The developer of the Security IC Embedded Software must ensure the appropriate “Treatment of User Data (A.Resp-Appl)” while developing this software in Phase 1 as specified below

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

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

This assumption requires that the Security IC 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 Security IC Embedded Software itself allows data to be compromised.

Examples of embedded software security concerns are given in section 7.2 of [BSI-PP-0035].

---

### 3.4.2 Assumptions from [PA]

#### **A.Key-Function** Usage of Key-dependent Functions

Key-dependent functions (if any) shall be implemented in the Security IC 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 Security IC 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 Security IC 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 that may be implemented in the Security IC Embedded Software. This uses the modular arithmetic functions of the MMC. 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 in which the embedded software uses the hardware. The properties of the TOE are assessed under this Security Target, but the software implementation is 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].

### 3.4.3 Other Assumptions

A variety of keys and other security-critical data may be injected for use by Security IC 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.

## 4. Security Objectives

### 4.1 Security Objectives for the TOE

#### 4.1.1 Objectives from [BSI-PP-0035]

The TOE shares the following high-level security goals from section 4.1 of [BSI-PP-0035]:

- SG1 maintain the integrity of User Data and of the Security IC Embedded Software (when being executed/processed and when being stored in the TOE's memories).
- SG2 maintain the confidentiality of User Data and of the Security IC Embedded Software (when being processed and when being stored in the TOE's memories).
- SG3 maintain the correct operation of the security services provided by the TOE for the Security IC Embedded Software.
- SG4 provide random numbers.

These high-level security goals in the context of the security problem definition build the standing point for the definition of security objectives as required by the Common Criteria. Note that the integrity of the TOE is a means to reach these objectives.

#### 4.1.1.1 Standard Security Objectives

##### **O.Leak-Inherent** Protection against Inherent Information Leakage

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

- by measurement and analysis of the shape and amplitude of signals (for example on the power, clock, or I/O lines)
- by measurement and analysis of 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. Details correspond to an analysis of attack scenarios which is not given here.

Note that this objective relates to security services provided by the TOE itself, and Security IC Embedded Software should ensure that the security services are appropriately used in conjunction with any additional leakage countermeasures implemented in software (cf. A.Plat-Appl and A.Resp-Appl in section 3.4.1).

##### **O.Phys-Probing** Protection against Physical Probing

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

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

with a prior

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

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

### **O.Malfunction**

Protection against Malfunctions

The TOE must ensure its correct operation.

The TOE must indicate or prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent malfunctions. Examples of environmental conditions are voltage, clock frequency, temperature, or external energy fields.

A malfunction of the TOE may also be caused using a direct interaction with elements on the chip surface. This is considered as being a manipulation (refer to the objective O.Phys-Manipulation) provided that detailed knowledge about the TOE's internal construction is required and the attack is performed in a controlled manner.

### **O.Phys-Manipulation**

Protection against Physical Manipulation

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

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

The TOE must be designed and fabricated so that it requires a high combination of complex equipment, knowledge, 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 Security IC must be protected against disclosure of confidential data processed in the Security IC (using methods as described under O.Leak-Inherent) even if the information leakage is not inherent but caused by the attacker

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

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

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

- (i) disclose critical User Data
- (ii) manipulate critical User Data of the Security IC Embedded Software
- (iii) manipulate Soft-coded Security IC Embedded Software
- (iv) bypass, deactivate, change or explore security features or security services of the TOE

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

**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 1.4.4.2.

**4.1.1.2 Security Objectives Related to Specific Functionality (referring to SG4)****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 a sufficient entropy.

---

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

#### 4.1.2 Objectives Based on [PA]

This section includes an objective for the TOE to provide a DES and a 3DES function, which is based on O.Add-Functions in [PA]. The modular arithmetic functions that are the basis for implementation of RSA and other asymmetric cryptographic algorithms in Security IC Embedded Software are part of the TOE functionality, but because they do not directly implement RSA (or other asymmetric systems), they are not included as cryptographic functions here.

##### **O.Add-Functions** Additional Specific Security Functionality

The TOE must provide the following specific security functionality to the Security IC Embedded Software:

- Triple Data Encryption Standard (3DES)

#### 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 SC5 in section 3.2.2.

SG5 software should be given the ability to respond to attacks.

This leads to objectives that

##### **O.SWResponse** Response to potential attacks by Security IC Embedded Software

The TOE shall allow Security IC Embedded Software to:

- (i) 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 and stop execution, or take some other action, if it detects a potential attack or other dangerous condition.

## 4.2 Security Objectives for the Environment

### 4.2.1 Security objectives for the security IC Embedded software development environment from [BSI-PP-0035]

#### Phase 1

---

**OE.Plat-Appl** Usage of Hardware Platform

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

- (i) hardware data sheet for the TOE:  
i.e. The TOE hardware manual [HM], and user guidance manual [UGM].
- (ii) data sheet of the IC Dedicated Software of the TOE,
- (iii) TOE application notes
- (iv) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as referenced in the certification report.

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

If required, the Security IC Embedded Software shall use these cryptographic services of the TOE and their interface as specified. When key-dependent functions implemented in the Security IC Embedded Software are just being executed, the Security IC Embedded Software must provide protection against disclosure of confidential data (User Data) stored and/or processed in the TOE by using the methods described under “Inherent Information Leakage (T.Leak-Inherent)” and “Forced Information Leakage (T.Leak-Forced)”.

**OE.Resp-Appl** Treatment of User Data

Security relevant User Data (especially cryptographic keys) are treated by the Security IC Embedded Software as required by the security needs of the specific application context.

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

By definition cipher or plain text data and cryptographic keys are User Data. The Security IC 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.

---

## 4.2.2 Security Objectives for the Operational Environment from [BSI-PP-0035]

### TOE Delivery up to the end of Phase 6

Appropriate “Protection during Packaging, Finishing and Personalisation (OE.Process-Sec-IC)” must be ensured after TOE Delivery up to the end of Phases 6, as well as during the delivery to Phase 7 as specified below.

#### **OE.Process-Sec-IC** Protection during composite product manufacturing

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

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-Sec-IC (which is the source of this objective) are given in section 3.1 of [BSI-PP-0035].

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

## 4.2.3 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 1.4.4.2. 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.

## 4.3 Security Objectives Rationale

The way in which [BSI-PP-0035] assumptions, organisational security policy and threats are met by objectives is given in section 4.4 of [BSI-PP-0035]. The table below includes the mapping from section 4.4 of [BSI-PP-0035] and adds the rationale for the additional assumptions, policy and threats in this Security Target.

**Table 4-1: Coverage of Security Assumptions, Policies and Threats by Objectives**

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

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 4.4 of [BSI-PP-0035] this applies to Phase 1 of the lifecycle.)

A.Process-Sec-IC is enforced by a directly corresponding requirement on the environment in OE.Process-Sec-IC. (Note that as in section 4.4 of [BSI-PP-0035] 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 4.4 of [BSI-PP-0035] this applies to Phase 1 of the lifecycle.)

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

The justification related to the organisational security policy "Protection during TOE Development and Production (P.Process-TOE)" is as follows:

O.Identification requires that the TOE has to support the possibility of a unique identification. The unique identification can be stored on the TOE. Since the unique identification is generated by the production environment the production environment must support the integrity of the generated unique identification. The technical and organisational security measures that ensure the security of the development environment and production environment are evaluated based on the assurance measures that are part of the evaluation. For a list of material produced and processed by the TOE Manufacturer refer to section 3.1 of [BSI-PP-0035]. All listed items and the associated development and production environments are subject of the evaluation.

Therefore, the organisational security policy P.Process-TOE is covered by this objective, as far as organisational measures are concerned.

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 4.4 of [BSI-PP-0035]: 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-0035] a clarification has been made for the security objective “Usage of Hardware Platform (OE.Plat-Appl)”: If required the Security IC Embedded Software shall use these cryptographic services of the TOE and their interface as specified. In addition, the Security IC 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-0035] 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 Security IC 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-0035] for the assumptions, policy and threats defined there.

Security IC Embedded Software may implement measures using the ability given by the TOE to embedded software to respond to the results of attacks based on these threats, in 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<sup>4</sup>. However, since no assumptions are made about the content of Security IC Embedded Software (and hence the use made of these features), these objectives are not included for the core threats in the table above.

T.NoSWResponse is directly addressed by O.SWResponse respectively.

T.RND is addressed by O.RND.

---

<sup>4</sup> 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.

## 5. Extended Components Definition

This ST does not define extended components and only refers to the extended components of [BSI-PP-0035].

### 5.1 Extended Components Definition from [BSI-PP-0035]

#### 5.1.1 Definition of the Family FCS\_RNG

The additional family (FCS\_RNG) of the Class FCS (cryptographic support) is defined in [BSI-PP-0035]. This family describes the functional requirements for random number generation used for cryptographic purposes.

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

#### 5.1.2 Definition of the Family FMT\_LIM

The additional family (FMT\_LIM) of the Class FMT (Security Management) is defined in [BSI-PP-0035]. This family describes the functional requirements for the Test Features of the TOE. The new functional requirements were defined in the class FMT because this class addresses the management of functions of the TSF. The examples of the technical mechanism used in the TOE appropriate to address the specific issues of preventing the abuse of functions by limiting the capabilities of the functions and by limiting their availability.

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

#### 5.1.3 Definition of the Family FAU\_SAS

The additional family (FAU\_SAS) of the Class FAU (Security Audit) is defined in [BSI-PP-0035]. This family describes the functional requirements for the storage of audit data. It has a more general approach than FAU\_GEN, because it does not necessarily require the data to be generated by the TOE itself and because it does not give specific details of the content of the audit records.

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

## 6. Security Requirements

### 6.1 Security Functional Requirements

The functional requirements for the TOE come from three sources:

- [BSI-PP-0035] These SFRs cover the core Security IC requirements:
  - FRU\_FLT.2
  - FPT\_FLS.1
  - FMT\_LIM.1
  - FMT\_LIM.2
  - FAU\_SAS.1
  - FPT\_PHP.3
  - FDP\_ITT.1
  - FPT\_ITT.1
  - FDP\_IFC.1
  - FCS\_RNG
- [PA] This SFR covers 3DES cryptographic requirements.
  - 3DES – FCS\_COP.1
- TOE features Some SFRs introduced from [BSI-PP-0035] 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 the ability of embedded software to cause a hardware reset (this is covered under FPT\_FLS.1 in section 6.1.1.1).

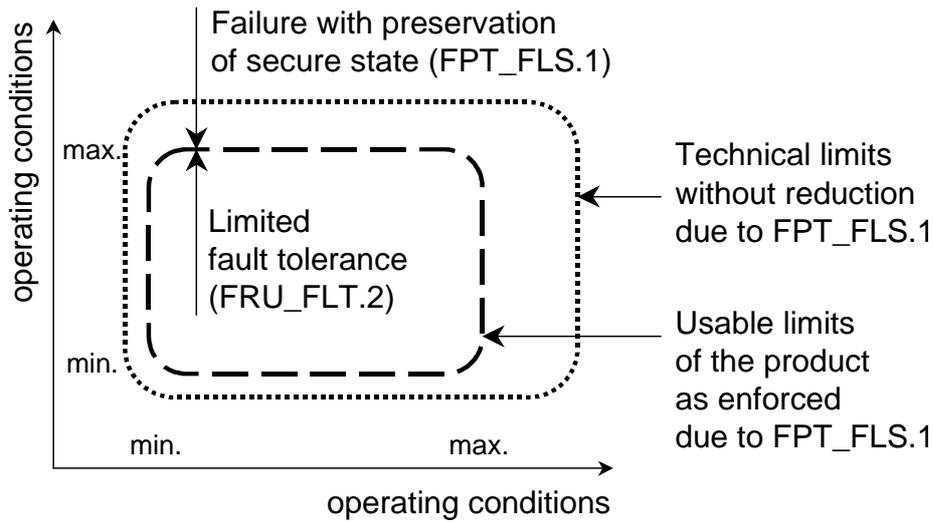
Note that all SFRs are drawn from [CC/2] except for FCS\_RNG.1, FMT\_LIM.1 & 2, and FAU\_SAS.1, which are all defined in section 5 of [BSI-PP-0035].

#### 6.1.1 Security Functional Requirements from [BSI-PP-0035]

In the specifications of SFRs listed below, 'Refinement' sections are taken from [BSI-PP-0035]; 'Application Notes' add information specific to the TOE.

##### 6.1.1.1 Prevention of Malfunction

The TOE implements a pair of security functional requirements that ensure it operates within conditions under which it can maintain a secure state. The secure state is represented in [BSI-PP-0035, fig 15], reproduced below:



**Figure 6-1: 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 Degraded fault tolerance

FRU\_FLT.2.1 The TSF shall ensure the operation of all the TOE’s capabilities when the following failures occur: *exposure to operating conditions which are not detected according to the requirement Failure with preservation of secure state (FPT\_FLS.1).*

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 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: No dependencies

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

The refinement above is defined in [BSI-PP-0035]. 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 make an illegal access
  - (ii) Attempts to execute an illegal instruction code
  - (iii) Processor halted by Security IC Embedded Software

### 6.1.1.2 Protection against Abuse of Functionality

The TOE controls access to test mode. Following [BSI-PP-0035], this is specified using the extended functional family FMT\_LIM, defined in section 5.2 of [BSI-PP-0035].

#### **FMT\_LIM.1 Limited capabilities**

Hierarchical to: No other components

FMT\_LIM.1.1 The TSF shall be designed and implemented in a manner that limits 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 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, 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 “availability” referred to in FMT\_LIM.2 are the functions implemented in the IC Dedicated Test Software.

The TOE stores identification/pre-personalisation data as described in section 1.4.4.2. This is included in [BSI-PP-0035] as the CC Part 2 extended functional component FAU\_SAS.1, replacing FAU\_GEN.1, and defined in section 6.1 of [BSI-PP-0035].

**FAU\_SAS.1**                      **Audit storage**

Hierarchical to:                      No other components

FAU\_SAS.1.1                      The TSF shall provide *the test process before TOE Delivery* with the capability to store *the Initialisation Data and/or Pre-personalisation Data and/or supplements of the Security IC Embedded Software* in *EEPROM*.

Dependencies:                      No dependencies

Application notes:

1. The data covered by “*Initialisation Data and/or Pre-personalisation Data and/or supplements of the Security IC Embedded Software*” is as identified in section 1.4.4.2.

The function Audit storage (FAU\_SAS.1) is subject to the limitations as specified by Limited availability (FMT\_LIM.2) above.

**6.1.1.3**                      **Protection against Physical Manipulation and Probing****FPT\_PHP.3**                      **Resistance to physical attack**

Hierarchical to:                      No other components

Dependencies:                      No dependencies

FPT\_PHP.3.1                      The TSF shall resist *physical manipulation and physical probing* to the *TSF* by responding automatically such that the SFRs are always enforced.

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

Application notes:

1. The TOE provides the reset state. When the TOE detects an illegal access, TOE will go to reset state automatically.

**6.1.1.4**                      **Protection against Leakage****FDP\_ITT.1**                      **Basic internal transfer protection**

Hierarchical to:                      No other components

---

FDP_ITT.1.1	The TSF shall enforce the <i>Data Processing Policy</i> to prevent the <i>disclosure</i> 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**

Hierarchical to:	No other components
FPT_ITT.1.1	The TSF shall protect TSF data from <i>disclosure</i> 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**

Hierarchical to:	No other components
FDP_IFC.1.1	The TSF shall enforce the <i>Data Processing Policy</i> on <i>all confidential data when they are processed or transferred by the TOE or by the Security IC Embedded Software</i> .
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 Security IC 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 Security IC Embedded Software.

#### **6.1.1.5 Generation of Random Numbers**

The TOE generates random numbers that can be used for cryptographic key generation. To capture the functional requirement, the family FCS\_RNG, defined in section 6.1 of [BSI-PP-0035] is used.

**FCS\_RNG.1 Random number generation**

Hierarchical to: No other components

FCS\_RNG.1.1 The TSF shall provide a *physical* random number generator that implements *total failure test of the random source [assignment: list of additional security capabilities]*

*Here, assignment : On-line test*

FCS\_RNG.1.2 The TSF shall provide random numbers that meet [*selection: independent bits with Shannon entropy of 7.976 bit per octet, Min-entropy of 7.95 bit per octet, [assignment: other comparable quality metric]*].

*Here, selection : independent bits with Shannon entropy of 7.976 bit per octet*

Dependencies: No dependencies.

**6.1.2 Security Functional Requirements Based on [PA]****6.1.2.1 Cryptographic Support**

The TOE provides a DES coprocessor. The DES coprocessor can be used to implement single or triple DES.

**FCS\_COP.1 Cryptographic operation**

FCS\_COP.1 is iterated here to address 3DES encryption and decryption (FCS\_COP.1 [3DES]).

**3DES Encryption and Decryption**

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 or 168 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, keying option 1 and 2*

Dependencies: [FDP\_ITC.1 Import of user data without security attributes, or FDP\_ITC.2 Import of user data without security attributes, or FCS\_CKM.1 Cryptographic key generation],  
FCS\_CKM.4 Cryptographic key destruction.

### Modular Arithmetic for Asymmetric Encryption and Decryption

The TOE also provides a modular multiplication coprocessor (see [HM, 16]) which can be used for the implementation of RSA (or other asymmetric algorithms) in Security IC Embedded Software. This can be used to build other FCS\_COP functions in software.

Because the TOE provides this generic functionality, rather than a specific cryptographic function such as RSA encryption or signature, the coprocessor functionality is covered under FRU\_FLT.2.

## 6.2 Security Assurance Requirements

The evaluation assurance level is EAL 5 augmented. An assurance level of EAL5 is required for smartcard product of TOE since it is intended to defend against highly sophisticated attacks without a protected environment. This evaluation assurance level was selected since it provides even formal evidence on the conducted vulnerability assessment. Table 6-1 describes the security assurance requirements. The increase of the assurance components compared [BSI-PP-0035] is expressed with bold letters. And the increase of the assurance components compared EAL5 are ALC\_DVS.2 and AVA\_VAN.5.

**Table 6-1: Assurance Components**

Assurance Class	Assurance components	Required by
ADV: Development	ADV_ARC.1 Security architecture description <b>ADV_FSP.5 Complete semi-formal functional specification with additional error information</b> ADV_IMP.1 Implementation representation of the TSF <b>ADV_INT.2 Well-structured internals</b> ADV_TDS.4 Semiformal modular design	EAL5, PP EAL5 EAL5, PP EAL5 EAL5
AGD: Guidance documents	AGD_OPE.1 Operational user guidance AGD_PRE.1 Preparative procedures	EAL5, PP EAL5, PP
ALC: Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation <b>ALC_CMS.5 Development tools CM coverage</b> ALC_DEL.1 Delivery procedures ALC_DVS.2 Identification of security measures ALC_LCD.1 Developer defined life-cycle model <b>ALC_TAT.2 Compliance with implementation standards</b>	EAL5, PP EAL5 EAL5, PP PP EAL5 EAL5
ATE: Tests	ATE_COV.2 Analysis of coverage <b>ATE_DPT.3 Testing: modular design</b> ATE_FUN.1 Functional testing ATE_IND.2 Independent testing - sample	EAL5, PP EAL5 EAL5 EAL5
AVA: Vulnerability assessment	AVA_VAN.5 Methodical vulnerability analysis	PP

Regarding, AVA\_VAN.5, the following Mandatory Technical Document is expected to be used for the vulnerability analysis:

Supporting Document, Mandatory Technical Document: Application of Attack Potential to Smartcards, April 2008, Version 2.5, Revision 1, CCDB-2008-04-001.

### 6.2.1 Refinements of the TOE Security Assurance Requirements

This ST claims conformance to the [BSI-PP-0035], and therefore it has to be conform to the refinements of the TOE security assurance requirements. Because the refinements in the PP are defined for the security assurance components of EAL4, some refinements have to be applied to assurance components of the higher level EAL5 stated in the Security Target.

Table 6-2 lists the influences of the refinements of the PP on the ST. Most of the refined security assurance components have the same level in both documents (PP and ST). The following two subsections apply the refinements to ALC\_CMS.5 and ADV\_FSP. 5 which are different between the PP and the ST.

**Table 6-2: Security Assurance Requirements, overview of differences of refinements**

Refined in PP	Influence on ST
ALC_DEL.1	Same as in PP, refinement valid without change
ALC_DVS.2	Same as in PP, refinement valid without change
ALC_CMS.4	ALC_CMS.5, refinements have to be adapted
ALC_CMC.4	Same as in PP, refinement valid without change
ADV_ARC.1	Same as in PP, refinement valid without change
ADV_FSP.4	ADV_FSP.5, refinements have to be adapted
ADV_IMP.1	Same as in PP, refinement valid without change
ATE_COV.2	Same as in PP, refinement valid without change
AGD_OPE.1	Same as in PP, refinement valid without change
AGD_PRE.1	Same as in PP, refinement valid without change
AVA_VAN.5	Same as in PP, refinement valid without change

### 6.2.2 Refinements regarding CM scope (ALC\_CMS)

This Security Target requires a higher evaluation level for the CC family ALC\_CMS, namely ALC\_CMS.5 instead of ALC\_CMS.4. The refinement of the PP regarding ALC\_CMS.4 is a clarification of the configuration items. Since in ALC\_CMS.5, the content and presentation of evidence element ALC\_CMS.5.1C only adds a further configuration item (development tool) to the list of items to be tracked by the CM system, the refinement can be applied without changes. The refinement of the configuration item of ALC\_CMS.4 can be found in section 6.2.1.3 of the [BSI-PP-0035] and is not cited here.

### 6.2.3 Functional specification (ADV\_FSP)

This ST requires a higher evaluation level for the CC family ADV\_FSP, namely ADV\_FSP.5 instead of ADV\_FSP.4. The refinement of the PP regarding ADV\_FSP.4 is concerned with the description of the TSF and its external interfaces, the purpose and method of use of all external TSF interfaces, the complete representation of the TSF and the accuracy and completeness of the TOE SFR instantiations. The refinement is not a change in the wording of the action elements, but a more detailed definition of the above items.

Since the higher level ADV\_FSP.5 requires a Functional Specification. The changes only affect the style of description and “error message” should be added in the functional specification. The refinements can be applied without changes and are valid for ADV\_FSP.5. The refinement of

---

the original component ADV\_FSP.4 can be found in section 6.2.1.6 of the [BSI-PP-0035] and is not cited here.

#### **6.2.4 Rationale for the Assurance Requirements**

ALC\_DVS.2 and AVA\_VAN.5, which have been augmented in 6.3.3 of [BSI-PP-0035], are additionally claimed in ST to comply with BSI-PP-0035.

Therefore, the information will be referred from [BSI-PP-0035].

##### **ALC\_DVS.2 Sufficiency of security measures**

Development security is concerned with physical, procedural, personnel and other technical measures that may be used in the development environment to protect the TOE.

In the particular case of a Security IC the TOE is developed and produced within a complex and distributed industrial process which must especially be protected.

Details about the implementation, (e.g. from design, test and development tools as well as Initialisation Data) may make such attacks easier. Therefore, in the case of a Security IC, maintaining the confidentiality of the design is very important.

This assurance component is a higher hierarchical component to EAL4 (which only requires ALC\_DVS.1). ALC\_DVS.2 has no dependencies.

##### **AVA\_VAN.5 Advanced methodical vulnerability analysis**

Due to the intended use of the TOE, it must be shown to be highly resistant to penetration attacks. This assurance requirement is achieved by the AVA\_VAN.5 component.

Independent vulnerability analysis is based on highly detailed technical information.

The main intent of the evaluator analysis is to determine that the TOE is resistant to penetration attacks performed by an attacker possessing high attack potential.

AVA\_VAN.5 has dependencies to ADV\_ARC.1 “Security architecture description”, ADV\_FSP.2 “Security enforcing functional specification”, ADV\_TDS.3 “Basic modular design”, ADV\_IMP.1 “Implementation representation of the TSF”, AGD\_OPE.1

“Operational user guidance”, and AGD\_PRE.1 “Preparative procedures”.

All these dependencies are satisfied by EAL4.

It has to be assumed that attackers with high attack potential try to attack Security ICs like smart cards used for digital signature applications or payment systems.

Therefore, specifically AVA\_VAN.5 was chosen in order to assure that even these attackers cannot successfully attack the TOE.

## 6.3 Security Requirement Rationale

### 6.3.1 Rational for the Security Functional Requirements

The way in which [BSI-PP-0035] objectives are implemented by SFRs is given in section 6.3 of [BSI-PP-0035]. The table below includes the mapping from section 6.3 of [BSI-PP-0035] and adds the rationale for the additional SFRs in this Security Target.

**Table 6-3: Security Requirements versus Security Objectives**

Objective	TOE Security Functional and Assurance Requirements
O.Leak-Inherent	<ul style="list-style-type: none"> <li>– FDP_ITT.1 “Basic internal transfer protection”,</li> <li>– FPT_ITT.1 “Basic internal TSF data transfer protection”,</li> <li>– FDP_IFC.1 “Subset information flow control”</li> </ul>
O.Phys-Probing	– FPT_PHP.3 “Resistance to physical attack”
O.Malfunction	<ul style="list-style-type: none"> <li>– FRU_FLT.2 “Limited fault tolerance”,</li> <li>– FPT_FLS.1 “Failure with preservation of secure state”</li> </ul>
O.Phys-Manipulation	FPT_PHP.3 “Resistance to physical attack”
O.Leak-Forced	All requirements listed for O.Leak-Inherent <ul style="list-style-type: none"> <li>– FDP_ITT.1, FPT_ITT.1, FDP_IFC.1,</li> </ul> plus those listed for O.Malfunction and O.Phys-Manipulation <ul style="list-style-type: none"> <li>– FRU_FLT.2, FPT_FLS.1, FPT_PHP.3</li> </ul>
O.Abuse-Func	<ul style="list-style-type: none"> <li>– FMT_LIM.1 “Limited capabilities”,</li> <li>– FMT_LIM.2 “Limited availability”,</li> </ul> plus those for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, <ul style="list-style-type: none"> <li>– FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, FPT_PHP.3,</li> <li>FRU_FLT.2, FPT_FLS.1</li> </ul>
O.Identification	FAU_SAS.1 “Audit storage”
O.RND	<ul style="list-style-type: none"> <li>– FCS_RNG.1 “Quality metric for random numbers”,</li> </ul> plus those for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation, O.Leak-Forced <ul style="list-style-type: none"> <li>– FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, FPT_PHP.3,</li> <li>FRU_FLT.2, FPT_FLS.1</li> </ul>
O.Add-Functions	FCS_COP.1 [3DES] “Cryptographic operation”
O.SWResponse	FPT_FLS.1
OE.Plat-Appl	not applicable
OE.Process-Sec-IC	not applicable
OE.Resp-Appl	not applicable
OE.InjDatSupp	not applicable

Reference is made to section 6.3 of [BSI-PP-0035] for the basic rationale. The remainder of this section deals with the additional parts of the rationale introduced for this Security Target.

O.Phys-Manipulation and O.Malfunction can be further addressed by Security IC 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 Security IC Embedded Software and is therefore beyond the scope of the TOE.

The 3DES requirement of O.Add-Functions is directly implemented by FCS\_COP.1.1[3DES].

---

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 Security IC 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. And more specifically by the security functional requirements specified below.

- [FDP\_ITC.1 Import of user data without security attributes, or FDP\_ITC.2 Import of user data without security attributes, or FCS\_CKM.1 Cryptographic key generation],
- FCS\_CKM.4 Cryptographic key destruction.

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

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 Security IC Embedded Software.

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-0035] for the assumptions, policy and threats defined there.

O.SWResponse is implemented by the ability of Security IC 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.

The assignment/selection operations performed on the SFRs drawn from [BSI-PP-0035] are shown in [BSI-PP-0035] itself. The additional operations performed in this ST are as follows:

**Table 6-4: Completion of SFRs**

<b>SFR</b>	<b>Operation required</b>	<b>Operation performed</b>
FAU_SAS.1	[assignment: list of subjects]	the test process before TOE Delivery
	[assignment: list of audit information]	the Initialisation Data and/or Pre-personalisation Data and/or supplements of the Security IC Embedded Software
	[assignment: type of persistent memory]	EEPROM
FCS_RNG.1	[assignment: list of additional security capabilities]	On-line test
	[selection: independent bits with Shannon entropy of 7.976 bit per octet, Min-entropy of 7.95 bit per octet, [assignment: other comparable quality metric]	independent bits with Shannon entropy of 7.976 bit per octet
FCS_COP.1 [3DES]	[assignment: list of cryptographic operations]	Encryption and decryption
	[assignment: cryptographic algorithm]	3DES
	[assignment: cryptographic key sizes]	112 or 168 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, keying option 1 and 2

### 6.3.2 Dependencies of Security Functional Requirements

The basic dependencies are shown in section 6.3.2 of [BSI-PP-0035] and are applicable to this ST – these are summarised in the table below:

**Table 6-5: Dependencies of Security Functional Requirements**

SFR	Dependencies	Fulfilled by Security Requirements in [BSI-PP-0035]?
FRU_FLT.2	FPT_FLS.1	Yes
FPT_FLS.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	FDP_ACC.1 or FDP_IFC.1	Yes
FDP_IFC.1	FDP_IFF.1	See discussion below
FPT_ITT.1	None	No dependency
FCS_RNG.1	None	No dependency

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

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

**Table 6-6: Additional SFR Dependencies**

SFR	Dependencies	Fulfilled by Security Requirements in this ST?
FCS_COP.1 [3DES]	[FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1] FCS_CKM.4	Yes (by the environment)
FDP_ITC.1	[FDP_ACC.1, or FDP_IFC.1] FMT_MSA.3	No additional requirement – see discussion below
FDP_ITC.2	[FDP_ACC.1, or FDP_IFC.1] [FPT_ITC, or FPT_TRP.1] FPT_TDC.	No additional requirement – see discussion below
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

The dependencies defined for FCS\_COP.1 in [CC/2] are discharged by 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, FDP\_ITC.2, FCS\_CKM.1, and FCS\_CKM.4 are not resolved because they will be fulfilled in the environment, where the appropriate decisions will be made.

---

The discussion in sections 6.3 and 7.3, and the rationale in section 6.3 of [BSI-PP-0035], 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.

## 7. TOE Summary Specification

### 7.1 TOE Security Functionalities

#### 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 RS45C to enter a reset state.

Correct operating ranges are defined in [HM].

#### 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.<sup>5</sup>

#### 3. 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 16-bit random numbers which satisfy the requirements of the monobit, poker, runs, long run, and autocorrelation tests in [AIS31].

#### 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. A suitable test routine is given in [UGM, 6].

#### 4. SF.DES

The TOE provides a hardware DES coprocessor that carries out 3DES encryption and decryption in ECB mode 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.  
(As to 3DES, keying option 1 and 2 of FIPS PUB 46-3 specifically)

---

<sup>5</sup> As with bus encryption, this protection is always active.

---

## Application Notes

1. The TOE provides a DES coprocessor, it is only actually used as an encryption and decryption function in the context of particular Security IC Embedded Software. For some application contexts, triple DES (as described in section 15.3.1 of [HM]) may be required in order to achieve a suitable strength of function.
  2. The TOE implements ECB mode - software can implement other modes such as CBC.
  3. 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, 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 Security IC Embedded Software is given in [UGM].
  4. The TOE provides a software DES library for RS-4. User should use DES library for RS-4 or should implement a function which is same level as DES library for RS-4 based on [UGM].
5. 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 Security IC Embedded Software to address threats to its correct operation by taking control of the operating environment. The Security IC 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.

- RAM Mirroring Function

When the RAMME bit of Detector Control Register (DTCR) is set by user software as described in section 17.3 of [HM], the RAM mirroring function is enabled, and data in a specific area in RAM (the area is specified in [HM]) is duplicated. An internal reset signal is issued, if any inconsistency occurs between the data in the area and the duplicated data while the RAM mirroring function is ON.

## 6. SF.TestModeControl

Once the TOE has been set to user mode, test mode functions are made not available.

The user mode is irreversibly set. It is however not impossible to transit back to test mode though that requires specific knowledge and highly sophisticated techniques.

## 7. SF.Inject

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

### The Reset State

The reset state is referred to in several of the TOE Security Functionalities 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]).

### HW.MMCCopro

The TOE provides a coprocessor that carries out modular multiplication according to the specification as in section 16 of [HM].

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

HW.MMCCopro refers to a critical function of the IC, but it is not specifically evaluated as a security function because the TOE provides generic modular arithmetic functionality, rather than a specific cryptographic function such as RSA encryption or signature.

## 7.2 Correspondence between TOE Security Functionalities and SFR

Table below shows the correspondence between SFRs and each of the Security Functionalities provided in the section 7.1 above.

**Table 7-1: TOE Security Functionalities Mapping to SFRs**

TOE Security Functionalities	SFR
SF.HWProtect,	FRU_FLT.2, FPT_FLS.1, FPT_PHP.3
SF.LeakProtect	FDP_ITT.1, FPT_ITT.1, FDP_IFC.1
SF.RNG	FCS_RNG.1
SF.DES	FCS_COP.1 [3DES]

TOE Security Functionalities	SFR
SF.ESFunctions,	FRU_FLT.2, FPT_FLS.1, FPT_PHP.3
SF.TestModeControl	FMT_LIM.1, FMT_LIM.2
SF.Inject	FAU_SAS.1

### 7.3 TOE Summary Specification Rationale

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

**Table 7-2: SFR Mapping to TOE Security Functionalities**

SFR	TOE Security Functionalities
FRU_FLT.2	SF.HWProtect, SF.ESFunctions
FPT_FLS.1	SF.HWProtect, SF.ESFunctions
FMT_LIM.1	SF.TestModeControl
FMT_LIM.2	SF.TestModeControl
FAU_SAS.1	SF.Inject
FPT_PHP.3	SF.HWProtect, SF.ESFunctions
FDP_ITT.1	SF.LeakProtect
FPT_ITT.1	SF.LeakProtect
FDP_IFC.1	SF.LeakProtect
FCS_RNG.1	SF.RNG
FCS_COP.1 [3DES]	SF.DES

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

## 8. Reference

### 8.1 Reference Materials

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

#### Literature

- [BSI-PP-0035] Security IC Platform Protection Profile, BSI-CC-PP-0035-2007, v1.0, Eurosmart, 15 June 2007
- [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
- [CC/1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, July 2009, version3.1, revision3, CCMB 2009-07-001
- [CC/2] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, July 2009, version3.1, revision3, CCMB 2009-07-002
- [CC/3] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, July 2009, version3.1, revision3, CCMB 2009-07-003

Note: the combination of all 3 parts 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

#### Document

- [HM] Renesas 16-bit Smart Card Microcomputer RS-4 Series RS45C Hardware Manual, Rev. 1.00, Renesas Technology Corp., 12 June, 2009
- [OPT] Option List for Smart Card Microcomputer (for RS45C), v.1.1, Renesas Technology Corp., 29 June, 2009
- [UGM] RS-4 Series User Guidance Manual, Rev. 1.30, Renesas Technology Corp., 6 August, 2009
- [SM] H8S/2600 Series H8S/2000 Series Software Manual FEB.24.06 Rev.4.00, Renesas Technology Corp., 24 February, 2006

### 8.2 Others

None.

\*\* End of Document \*\*