RS4FC128 Version 01

Security Target

-Public Version-

Renesas Electronics Corporation

Kenji Hirao

0. History

0.1 Approval

	Name	Date
Prepared	Kenji Hirao	26 September, 2013
Approved (1)	—	—
Approved (2)	Takashi Endo	26 September, 2013

0.2 Abbreviations

Term	Meaning
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining A mode of DES and AES 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 PKCC
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 AES encryption.
ECC	Elliptic Curve Cryptography
EEPROM	Electronically Erasable Programmable Read-Only Memory
HW RNG	Hardware random number generator (physical random number generator)
IC	Integrated Circuit
IT	Information Technology
MCU	Micro Computer Unit
PKCC	Public Key Cryptography Coprocessor
PKI	Public Key Infrastructure
PP	Protection Profile
OFB	Output Feedback A mode of DES and AES encryption.
OS	Operating System
RAM	Random Access Memory
RL	Random Logic (Glue Logic)
RNG	Random Number Generator
ROM	Read-Only Memory
RSA	Rivest, Shamir, Adelmann – a public key encryption algorithm, named after its inventors.
SFP	Security Function Policy
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality

0.3 Glossary

Term	Meaning
Embedded Software	Software held in the chip, having been developed by users of the TOE. Such software generally includes an operating system and may include all or part of applications. Embedded Software is held in EEPROM or RAM. The chip does not depend on such software, and Embedded Software is not part of the TOE. Note that Embedded Software includes all software on a TOE other than the IC Dedicated Software. Embedded Software is also referred to as 'Security IC Software' (especially in [BSI-PP-0035]).
EWE	An interrupt generated by the TOE whenever an attempt is made to write to EEPROM.
FMU	Firewall Management Unit – a feature of the TOE that limits the memory areas available.
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	A form supplied by Renesas and filled in by a TOE customer, specifying various options for the manufacture of TOE ICs for that customer. The aspect of particular interest to this security target is : • Selection of whether pre-personalisation data injection is required The option list also describes the content and structure of the manufacturing identification data that Renesas will inject (see section 1.4.4.2).
Renesas	Refer to Renesas Electronics Corporation (http://www.renesas.com/)
Reset state	A state in which the chip does not execute instructions or engage in input/output. The chip can exit the reset state by receiving an external reset. See also section 7.1.
Secure Boot Loader Software	Secure Boot Loader software from Renesas
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	(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. For the RS4FC128, the TOE Manufacturer refers to Renesas.
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.
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 2
      1.4.1 Product Description 4
      1.4.2 TOE Intended Usage 7
      1.4.3 TOE Lifecycle 7
      1.4.4 TOE Environments 10
2. Conformance Claims 13
   2.1 CC Conformance Claim 13
   2.2 PP Claims 13
      2.2.1 PP Reference 13
      2.2.2 PP Tailoring 13
      2.2.3 PP Additions 13
   2.3 Package Claim 13
   2.4 Conformance Rationale 13
      2.4.1 CC Conformance Rationale 13
      2.4.2 PP Claim Rationale 14
      2.4.3 Package Claims Rationale 14
3. Security Problem Definition 15
   3.1 Description of Assets 15
   3.2 Threats 16
      3.2.1 Threats Defined in [BSI-PP-0035] 16
      3.2.2 Other Threats 19
   3.3 Organisational Security Policies 20
      3.3.1 Policy Requirement from [BSI-PP-0035] 20
      3.3.2 Policy Requirement from [PA] 21
      3.3.3 Other Policy Requirements 21
   3.4 Assumptions 21
      3.4.1 Assumptions from [BSI-PP-0035] 21
      3.4.2 Assumptions from [PA] 22
      3.4.3 Other Assumptions 23
4. Security Objectives 24
   4.1 Security Objectives for the TOE 24
      4.1.1 Objectives from [BSI-PP-0035] 24
      4.1.2 Objectives Based on [PA] 27
      4.1.3 Other Objectives 27
   4.2 Security Objectives for the Environment 29
      4.2.1 Security objectives for the security IC Embedded software development environment from [BSI-PP-0035] 29
      4.2.2 Security Objectives for the Operational Environment from [BSI-PP-0035] 30
      4.2.3 Other Environment Security Objectives 30
   4.3 Security Objectives Rationale 31
5. Extended Components Definition 34
   5.1 Extended Components Definition from [BSI-PP-0035] 34
      5.1.1 Definition of the Family FCS_RNG 34
      5.1.2 Definition of the Family FMT_LIM 34

5.1.3 Definition of the Family FAU_SAS ... 34
6. Security Requirements ... 35
6.1 Security Functional Requirements ... 35
6.1.1 Security Functional Requirements from [BSI-PP-0035] ... 36
6.1.2 Security Functional Requirements Based on [PA] ... 41
6.1.3 Security Functional Requirements from TOE features ... 43
6.2 Security Assurance Requirements ... 46
6.2.1 Refinements of the TOE Security Assurance Requirements ... 47
6.2.2 Refinements regarding CM scope (ALC_CMS) ... 47
6.2.3 Functional specification (ADV_FSP) ... 47
6.2.4 Rationale for the Assurance Requirements ... 48
6.3 Security Requirement Rationale ... 49
6.3.1 Rational for the Security Functional Requirements ... 49
6.3.2 Dependencies of Security Functional Requirements ... 55
7. TOE Summary Specification ... 57
7.1 TOE Security Functionalities ... 57
7.2 Correspondence between TOE Security Functionalities and SFR ... 61
7.3 TOE Summary Specification Rationale ... 62
8. Reference ... 63
8.1 Reference Materials ... 63
8.2 Others ... 63

List of Figures

Figure 1-1: Configuration of the TOE ... 3
Figure 1-2: Internal Block Diagram of the TOE ... 4
Figure 1-3: Design and Manufacturing Lifecycle ... 8
Figure 1-4: Modes ... 10
Figure 6-1: Paradigm Regarding Operating Conditions ... 36

List of Tables

Table 1-1: Detail of product type ... 2
Table 1-2: TOE Configuration ... 2
Table 4-1: Coverage of Security Assumptions, Policies and Threats by Objectives ... 31
Table 6-1: Assurance Components ... 46
Table 6-2: Security Assurance Requirements, overview of differences of refinements ... 47
Table 6-3: Security Requirements versus Security Objectives ... 49
Table 6-4: Completion of SFRs ... 51
Table 6-5: Dependencies of Security Functional Requirements ... 55
Table 6-6: Additional SFR Dependencies ... 55
Table 7-1: TOE Security Functionalities Mapping to SFRs ... 61
Table 7-2: SFR Mapping to TOE Security Functionalities ... 62

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:	RS4FC128 Version 01, Security Target
Revision:	$Rev:: 6655 $
Provided by:	Renesas Electronics Corporation

1.2 TOE Reference

Renesas RS4FC128 Version 01 integrated circuit Product Type Code 00, and

Renesas RS4FC128E Version 01 integrated circuit Product Type Code 01.

1.3 TOE Overview

This document is the ST for the Renesas RS4FC128 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.

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, e-m-commerce, digital signature, USIM/UMTS, and credit card.

Applications such as 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 Memory for program storage over previous devices whilst ensuring a balance of EEPROM for data storage.

1.4 TOE Description

RS4FC128 is the generic name of this device. But there are some sales name using this TOE is listed below. In this document, normally used RS4FC128 as TOE, but sometime used sales name as required. Table 1-1 is the list of products.

Table 1-1: Detail of product type

Type Name	Communication IF	Package at Shipment	Application	Product Type Code (detail is shown in [OPT])
RS4FC128	Contact	Wafer, COT	Cards	00
RS4FC128E	Contact	Wafer, SON8	Embedded in devices	01

The differences between RS4FC128 and RS4FC128E are shown in injection data (see [OPT]).

The TOE configuration is summarised in the table below:

Table 1-2: TOE Configuration

Item Type	Item	Version	Form of delivery
Hardware	RS4FC128 integrated circuit (refer to Table 1-1 for detail)	01	Wafer or packaged module (see section 1.4.3)
Software	IC Dedicated Test Software Test ROM software	50282	Included in RS4FC128 test ROM

Item Type	Item	Version	Form of delivery
Software	Secure Boot Loader software	5560	Included in RS4FC128 test ROM
Software	RNG on-line test software	1.1 (defined by the version of [UGM])	Hardcopy: provided as a part of [UGM]. (This is implemented in the Embedded Software by the user)
Document	Hardware Manual: [HM]	1.00	Electronic data/Hardcopy
Document	Secure Boot Loader Version 5560 User's Manual: [SBLM]	1.10	Electronic data/Hardcopy
Document	User Guidance: [UGM]	1.1	Electronic data/Hardcopy
Document	Option List: [OPT]	0.2	Electronic data/Hardcopy

Further description of the TOE is provided in section 1.4.1.

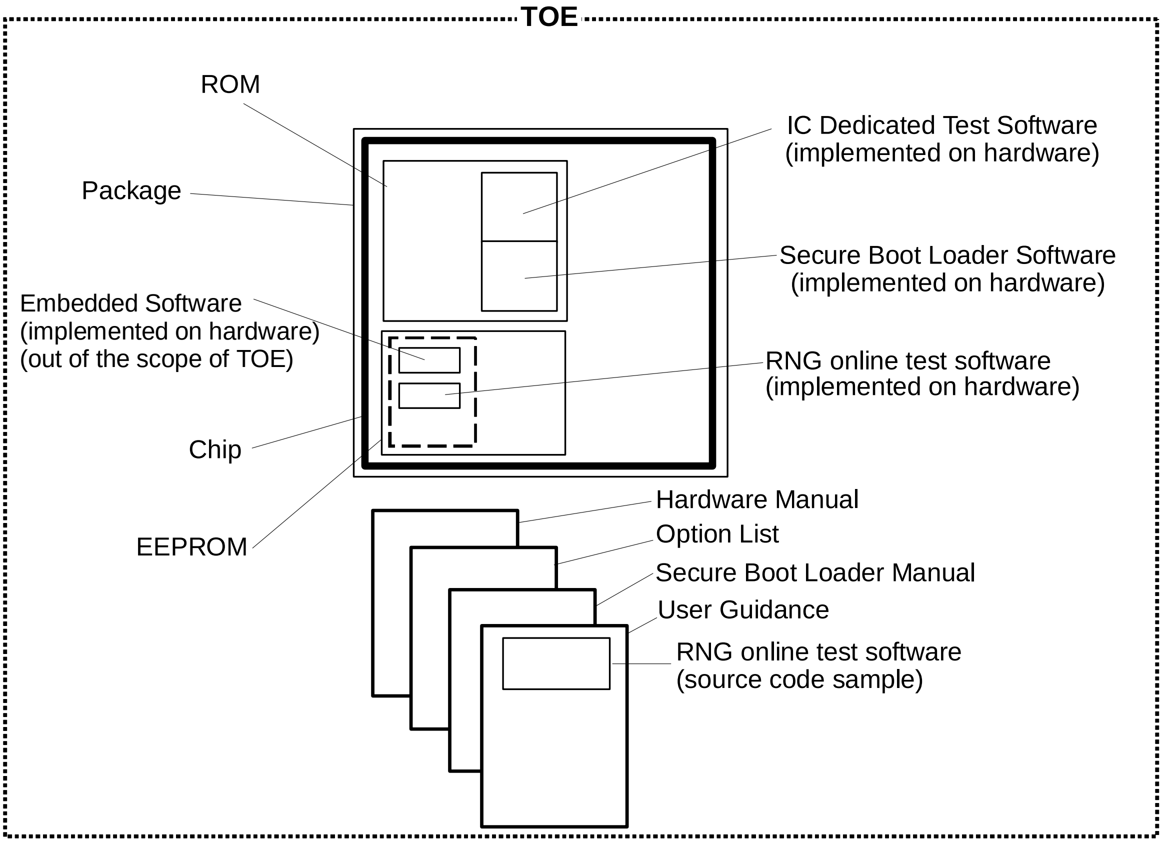
TOE
ROM
Package
Embedded Software
(implemented on hardware)
(out of the scope of TOE)
Chip
EEPROM
IC Dedicated Test Software
(implemented on hardware)
Secure Boot Loader Software
(implemented on hardware)
RNG online test software
(implemented on hardware)
Hardware Manual
Option List
Secure Boot Loader Manual
User Guidance
RNG online test software
(source code sample)

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.

1.4.1 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:

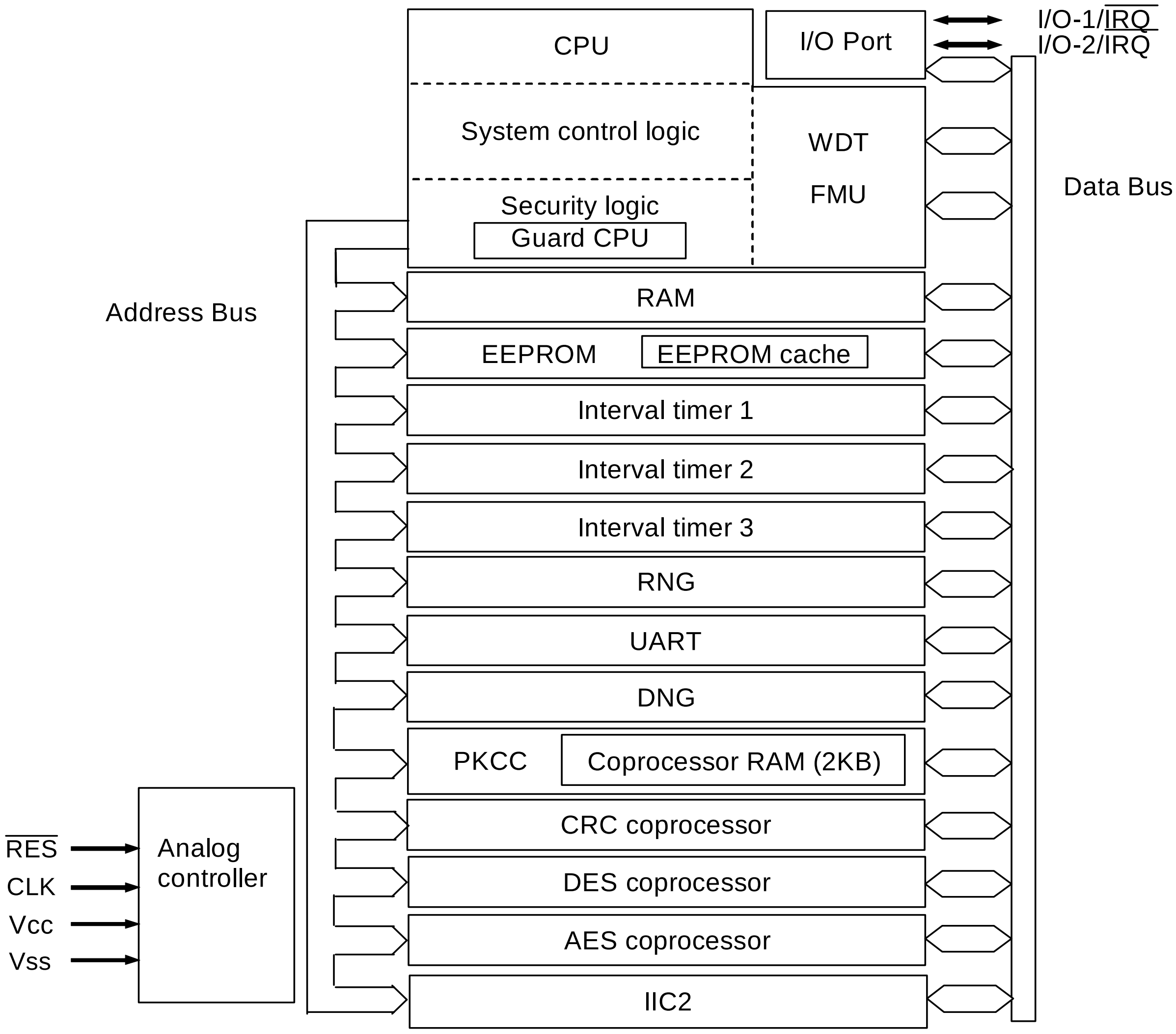

Figure 1-2: Internal Block Diagram of the TOE

1.4.1.1 Hardware

The TOE is an integrated circuit based around the high-speed CPU 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 include Guard CPU: ROM, RAM, EEPROM, random number generator (RNG), PKCC, AES coprocessor, DES coprocessor, CRC coprocessor, UART, three interval timers, WDT, FMU, IIC2, and two I/O lines.

CPU: the CPU can operate with any one of a 1.8V, 3V or 5V supply and a maximum internal clock frequency of 33MHz and is upwards compatible with Renesas' AE-4 series Security ICs.

- 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 (EEPMOV.B and EEPMOV/P.W) and high-speed block transfer instruction (EEPMOV.W)
- The Guard CPU has a function for monitoring the MOV instruction and Bcc instruction. The Guard CPU calculates the address and data of these instructions in parallel with the CPU and then compares the obtained result with the calculation result of the CPU. If the comparison result is not equal, a security reset is initiated.

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, which can be used as normal RAM when not required by the coprocessor.

- ROM: 64kbytes
    - 32 kbytes (for IC Dedicated Test Software)
    - 32 kbytes (for Secure Boot Loader Software)
- RAM: 6.5kbytes + 2 kbytes of coprocessor RAM (CRAM)
- EEPROM: 128kbytes
    - on-chip charge pump and independent oscillator
    - special write instruction, and interrupt generation on writing
    - page write/erase (128 bytes)

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.

PKCC (Public-key cryptographic coprocessor): the public key cryptography coprocessor features high-speed operations such as RSA cryptography power residue operation and ECC cryptography scalar multiplication of the points on an elliptic curve. *The only functionality of this coprocessor included in the evaluated hardware configuration as part of the TOE is* RSA encryption/decryption according section 5.1.1 RSAEP/5.1.2 RSADP in [RSA]. *Any use in cryptographic software other than described above is not included.*

AES coprocessor: The AES coprocessor module executes the AES encryption for ECB mode, CBC mode, and OFB mode in hardware. Different combination of key and data lengths can be used (128, 192 or 256 bits for key, 128 bits for data). 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.

DES coprocessor: this hardware engine can be used to provide either DES or triple-DES functions for (T)ECB mode, (T)CBC mode, and (T)OFB mode in hardware. 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. For some application contexts, Single-DES may be sufficient [1] - 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.

FMU: Three types of monitor functions are provided.

(i) Monitoring memory read/write

(ii) Monitoring execution of the program in the memory

(iii) Monitoring access to peripheral modules

I/O-1/IRQ, I/O-2/IRQ: 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 that conforms to the ISO/IEC standard 7816-3. Full-duplex asynchronous mode that conforms to the ISO/IEC CD standard 10536-4:

IIC2: The I2C bus interface 2 conforms to the NXP Semiconductors I2C bus (inter-IC bus) interface standard and provides a subset of the functions.

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.

Full details of the operation of the chip and guidance for its use are given in [HM], [OPT], and [UGM].

[1] 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.

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.

Secure Boot Loader Software:

The Secure Boot Loader software is integrated into the ROM of the TOE hardware. When the chip is first put in boot mode, the Secure Boot Loader Software is activated, which will download the Security IC embedded software. When the Secure Boot Loader downloads the user's security IC embedded software, the Secure Boot Loader software does an RSA verification and AES decryption for integrity and confidentiality of the user's security IC embedded software. RSA verification is done according section 5.2.2 in [RSA]. AES decryption uses the AES coprocessor. After a successful download of the security IC embedded software, the TOE reboots into normal mode and the user's security IC embedded software is called. Once the device is locked by the user, the Secure Boot Loader is disabled for further downloads.

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.

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 RS4FC128 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]. The Secure Boot Loader Manual [SBLM] is supplied at the basic reference for users who are developing Security IC Embedded Software.

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, 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.

There are two ways for embedded software to be loaded depending on customer preference: Either embedded software is provided to Renesas and loaded into EEPROM by Renesas, or embedded software is installed in EEPROM using the Secure Boot Loader at the customer's site. The details are shown below.

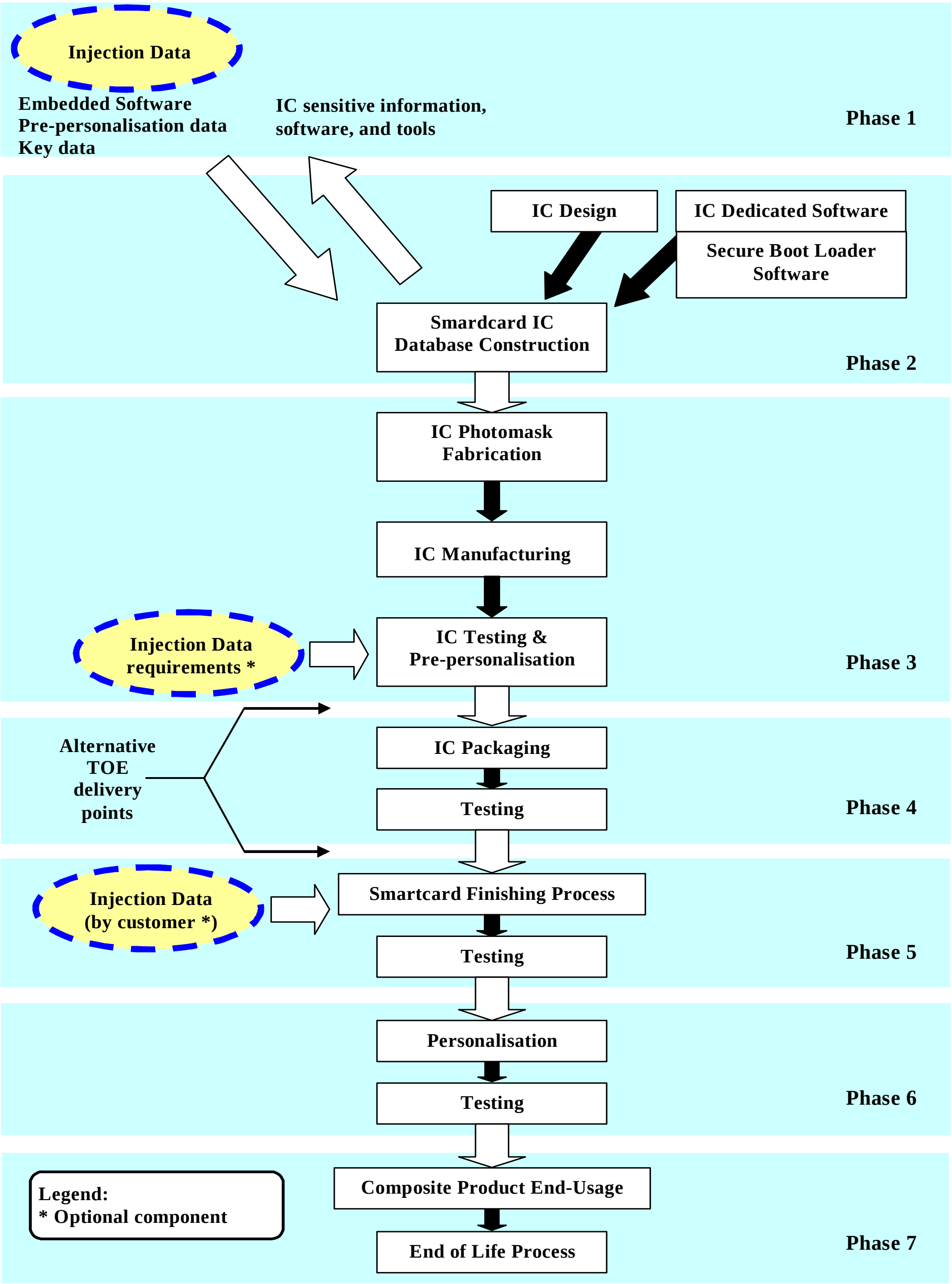
Injection Data
Embedded Software
Pre-personalisation data
Key data
IC sensitive information, software, and tools
Phase 1
IC Design
IC Dedicated Software
Secure Boot Loader Software
Smardcard IC Database Construction
Phase 2
IC Photomask Fabrication
IC Manufacturing
Injection Data requirements *
IC Testing & Pre-personalisation
Phase 3
Alternative TOE delivery points
IC Packaging
Testing
Phase 4
Injection Data (by customer *)
Smartcard Finishing Process
Testing
Phase 5
Personalisation
Testing
Phase 6
Legend:
* Optional component
Composite Product End-Usage
End of Life Process
Phase 7

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).

Any embedded software for incorporation in the EEPROM is sent via a secure delivery method from the software developer to Renesas, and its secure handling is then ensured by . If the secure boot loader is to be used, key 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 developed in this phase, to enable testing of the IC at various phases of its manufacture. The Secure Boot Loader Software is developed in this phase, to enable installing of the customer embedded software in phase 5 as required. The security IC embedded softwareis received from phase 1 if Renesas will install it in the EEPROM. Or, the key data is received from phase 1 to prepare for use of the Secure Boot Loader.

  The security of the development environments for the IC, its dedicated test software and the Secure Boot Loader 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. Also, in this phase, wafers containing the TOE are produced. If delivered to Renesas during phase 1, the Security IC Embedded Software will be installed via EEPROM programming during manufacturing, or, the key data for the Secure Boot Loader received in phase 1 is injected into EEPROM. 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 injection 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 Boot mode or Normal Mode). Alternatively, the TOE may be packaged by Renesas and delivered in COT/SON8 at the end of phase 4. If the TOE is to be delivered in COT/SON8 then at the end of phase 3 it is set to a constrained Test Mode, instead of Boot Mode or Normal 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 the customer's responsibility. Note that in all cases the TOE will be set to Boot Mode or Normal 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 Boot Mode or Normal.

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. During phase 5, the customer loads the Security IC Embedded Software into the EEPROM (if not provided to Renesas to load during phase 3).

1.4.3.1 Modes

RS4FC128 has three modes as shown in Figure 1-4.

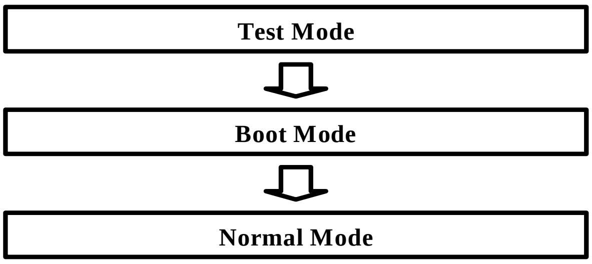

Figure 1-4: Modes

During manufacturing and production (phases 2-4), the chip makes 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.

- Boot mode makes the IC Dedicated Test Software unavailable and instead the function to download the Security IC Embedded Software into the EEPROM is available.

- Normal mode provides only the functionality described in [HM]. The software interface then 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 in Boot Mode or Normal Mode at the point of TOE Delivery depending on whether the Security IC Embedded Software was programmed during Phase 3. The transition from Boot Mode or Normal Mode back 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 Boot Mode or Normal Mode depending on whether the Security IC Embedded Software was programmed during Phase 3. 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 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 Secure Boot Loader Software
  - Confidentiality and integrity of customer EEPROM code and injection data (i.e. the Security IC Embedded Software for an instance of the TOE)
- Mask manufacture site
  - Confidentiality and integrity of design and base masks
- Manufacturing sites
  - Confidentiality and integrity of base masks
  - 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 EEPROM.

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.

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 or key data 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 EEPROM, an IC will also require software to be added after it leaves the manufacturing environment. Operating system software may require additional parts or patches to be loaded, and increasingly applications are expected to be loaded and deleted after a Composite Product embedded with a 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 the specification of EEPROM and the other security functions – 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 of 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[2]

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.

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

[2] 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.

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 Boot Mode or Normal 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.NoSWDetect Inability of the TOE to detect an attack

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

The TOE is concerned in particular to detect the following:

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

(ii) Attempts to make an illegal access

(iii) Attempts to execute code from a type of memory not permitted by the software environment[3]

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

(v) Attempts to execute an illegal instruction code

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

For the purposes of this threat, an illegal access is one that is marked "×" in the tables titled "Access" in the TOE memory map (section 4 of [HM]). In addition, among the accesses marked "○" in the tables, if a type of access allowed is noted under the circle, an access other than the allowed type of access is also an illegal access.

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

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

T.NoSWResponse Inability of 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[4].

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.

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

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)
- Advanced Encryption Standard (AES)
- R.Rivest, A.Shamir and L.Adleman (RSA)

3.3.3 Other Policy Requirements

As an additional policy, the TOE provides specific security functionality which provided by the Secure Boot Loader Software.

P.SWDownload Control of the software downloads Security Functionality

The TOE shall provide the capability to download customer's security IC embedded software into the EEPROM in secure way.

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).

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 PKCC. In this case the leakage characteristic of the implementation will depend partly on the hardware characteristics of the TOE and its PKCC, 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 3DES and an AES 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)
- Advanced Encryption Standard (AES)
- R.Rivest, A.Shamir and L.Adleman (RSA)

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 Other Threats.

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

This leads to objectives that

O.SWDetect Detection of potential attacks by the TOE

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

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

(ii) Attempts to make an illegal access

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

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

(v) Attempts to execute an illegal instruction code

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

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

O.SWResponse Response to potential attacks by 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.

O. SWDownload Control of the software downloads Security Functionality

The TOE will provide the capability to download customer's security IC embedded software into the EEPROM memory. Once the security IC embedded software has been download successfully and locked into the device, the Secure Boot Loader will be disabled for further downloads. The TOE must support confidentiality and integrity of user data and of the Security IC Embedded Software.

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

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.NoSWDetect	O.SWDetect
T.NoSWResponse	O.SWResponse
P.SWDownload	O.SWDownload

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.SWDetect and O.SWResponse. This can help address some of the core threats – T.Phys-Manipulation, T.Malfunction and T.Abuse-Func by detecting the results of attempts to tamper with the operation of the IC, and using additional defensive measures at the level of the target of the attack[6]. 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.NoSWDetect is directly addressed by O.SWDetect.

T.NoSWResponse is directly addressed by O.SWResponse.

T.RND is addressed by O.RND.

Since O.SWDownload requires the TOE to implement the same specific security functionality as required by P.SWDownload, the organisational security policy is covered by the objective.

[6] 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] according to [AIS31]. 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.1
- [PA] This SFR covers 3DES, AES and RSA cryptographic requirements.
  - 3DES – FCS_COP.1
  - AES – FCS_COP.1
  - RSA – 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).
  - Control of the software downloads – FCS_COP.1 [AES], FCS_COP.1 [RSA signature verification], FDP_ITC.1, FDP_ACC.1 [SBL], FDP_ACF.1 [SBL], FMT_MSA.3, FMT_MSA.1 and FM_SMF.1 are added to support integrity of user data and of the Security IC Embedded Software.

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 reset state makes the secure state for FPT_FLS.1. The secure state is represented in [BSI-PP-0035, fig 15], reproduced below:

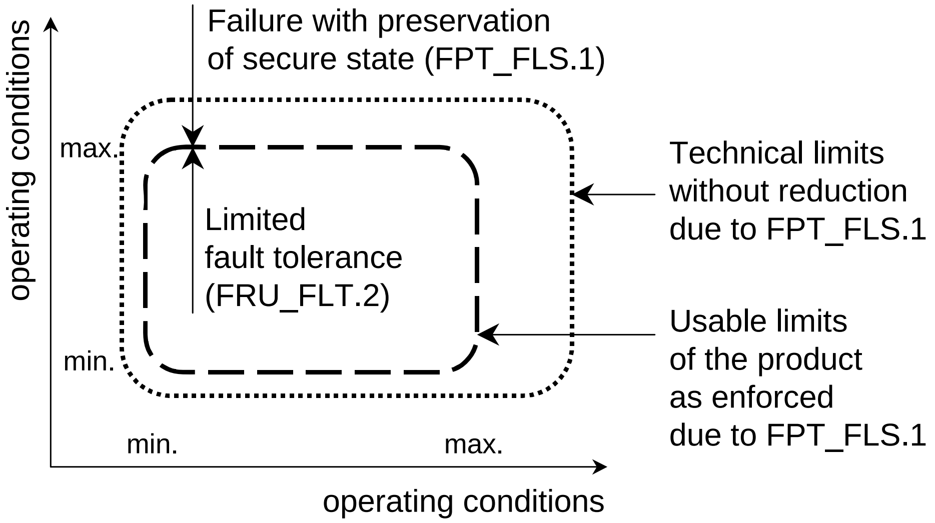

Figure 6-1: Paradigm Regarding Operating Conditions

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 is 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 *Data Processing Policy* to prevent the *disclosure* of user data when it is transmitted between physically-separated parts of the TOE.

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

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

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

FPT_ITT.1 **Basic internal TSF data transfer protection**

Hierarchical to: No other components

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

Dependencies: No dependencies.

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

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

FDP_IFC.1 **Subset information flow control**

Hierarchical to: No other components

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

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. The family "Generation of Random Numbers (FCS_RNG.1)" has to be newly created according [AIS31]. This security functional component is used instead of the functional component FCS_RNG.1 defined in [BSI-PP-0035].

FCS_RNG.1 Random number generation (Class PTG.2)

Hierarchical to: No other components

FCS_RNG.1.1 The TSF shall provide a *physical*[7] random number generator that implements:

(PTG.2.1) *A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.*

(PTG.2.2) *If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source*[8]*.*

(PTG.2.3) *The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.*

(PTG.2.4) *The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.*

(PTG.2.5) *The online test procedure checks the quality of the raw random number sequence. It is triggered applied upon specified internal events: start-up, before and after key generation*[9]*. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.*[10]

FCS_RNG.1.2 The TSF shall provide *numbers in the format 16-bit* that meet:

(PTG.2.6) *Test procedure A, as defined in [AIS31] does not distinguish the internal random numbers from output sequences of an ideal RNG.*

(PTG.2.7) *The average Shannon entropy per internal random bit exceeds 0.997.* [11]

[7] [selection: *physical*]

[8] [selection: *prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source*]

[9] [selection: *applied upon specified internal events: start-up, before and after key generation*]

[10] [assignment: *list of security capabilities*]

[11] [assignment: *a defined quality metric*]

Dependencies: No dependencies.

6.1.2 Security Functional Requirements Based on [PA]

6.1.2.1 Cryptographic Support

The TOE provides a DES, an AES and a PKCC (RSA) 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]), AES encryption and decryption (FCS_COP.1 [AES]) and RSA encryption and decryption (FCS_COP.1 [RSA Encryption and Decryption]). And also RSA verification (FCS_COP.1 [RSA signature verification]) is addressed separately.

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) in one of the following modes of operation: ECB, CBC, OFB,* and cryptographic key sizes *of 112 or 168 bits* that meet the following *standards*:

Technology Administration U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm(TDEA) Block Cipher, Revised 19 May 2008, NIST Special Publication 800-67, Version 1.1

'DES MODES OF OPERATION' Federal Information Processing Standards Publication 81, 2nd December 1980 (http://www.itl.nist.gov/fipspubs/fip81.htm)

National Institute of Standards and Technology (NIST), Technology Administration, U.S. Department of Data Encryption Standard (DES), NIST Special Publication 800-38A, 2001 Edition.

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.

Application Notes:

1. For the case of 2-key 3DES, the Cryptographic functionality FCS_COP.1 [3DES] provided by the TOE achieves a security level no more than 80 bits.

AES Encryption and Decryption

FCS_COP.1 [AES]

Hierarchical to: No other components

FCS_COP.1.1 [AES] The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *Advanced Encryption Standard (AES) in one of the following modes of operation: ECB, CBC, OFB,* and cryptographic key sizes of *128, 192, and 256 bits* that meet the following *standards*:

U.S. Department of Commerce / National Bureau of Standards Advanced Encryption Standard, FIPS PUB 197, 2001 November 26. National Institute of Standards and Technology Special Publication 800-38A 2001 Edition, Recommendation for Block Cipher Modes of Operation Methods and Techniques.

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.

Application Notes:

1. For all key sizes, the Cryptographic functionality FCS_COP.1 [AES] provided by the TOE achieves a security level in excess of 80 bits.

RSA Encryption and Decryption

FCS_COP.1 [RSA Encryption and Decryption]

Hierarchical to: No other components

FCS_COP.1.1 [RSA Encryption and Decryption]

The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *R.Rivest, A.Shamir and L.Adleman (RSA)* and cryptographic key sizes *of 1024-2176 bits* that meet the following *standards*:

Encryption: Section 5.1.1 RSAEP in [RSA].

Decryption (without CRT): Section 5.1.2 RSADP in [RSA].

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.

Application Notes:

1. For the case of 1024 bit RSA, the Cryptographic functionality FCS_COP.1 [RSA Encryption and Decryption] provided by the TOE achieves a security level no more than 80 bits.

<u>RSA Signature verification</u>

FCS_COP.1 [RSA signature verification]

Hierarchical to: No other components

FCS_COP.1.1 [RSA signature verification]

The TSF shall perform *verification* in accordance with a specified cryptographic algorithm *R.Rivest, A.Shamir and L.Adleman (RSA)* and cryptographic key size *of 2048 bits* that meet the following *standards*:

Verification: Section 5.2.2 RSAVP1 in [RSA].

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.

<u>Public Key Cryptography Coprocessor for Encryption and Decryption</u>

The public key cryptography coprocessor of TOE provides high-speed operations such as RSA cryptography power residue operation and ECC cryptography scalar multiplication of the points on an elliptic curve for Security IC Embedded Software.

6.1.3 Security Functional Requirements from TOE features

FDP_ACC.1 Subset access control

FDP_ACC.1 is iterated here to address the access control by the Secure Boot Loader software (FDP_ACC.1 [SBL]).

FDP_ACC.1 [SBL]

Hierarchical to: No other components

FDP_ACC.1.1 [SBL] The TSF shall enforce the *Control of software download Policy* on *the execution of Secure Boot Loader software instructions.*

Here,
Policy: Control of software downloads Policy
Subjects: Secure Boot Loader software
Objects: instruction
Operations: execution

Control of Software download Policy

Only the IC Embedded software with correct signature can be downloaded.

Dependencies: FDP_ACF.1 [SBL] Security attribute based access control

FDP_ACF.1 Security attribute based access control

FDP_ACF.1 is iterated here to address the security attribute based access control by the Secure Boot Loader software (FDP_ACF.1 [SBL]).

FDP_ACF.1 [SBL]

Hierarchical to: No other components

FDP_ACF.1.1 [SBL][12] The TSF shall enforce the *Control of software download Policy* to objects based on the following:

Subject: Secure Boot Loader software
Object: the IC Embedded Software
attributes: correct signature

FDP_ACF.1.2 [SBL] The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *the Security Boot Loader Software can be executed downloading of the IC Embedded Software only, if the signature verification process is succeeded.*

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

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

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

FDP_ITC.1 Import of user data without security attributes

Hierarchical to: No other components

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

FDP_ITC.1.1 The TSF shall enforce the *Control of Software download Policy* when importing user data, controlled under the SFP, from outside of the TOE.

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

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: *none*.

Dependencies: [FDP_ACC.1 subset access control, or FDP_IFC.1 Subset information flow control]
FMT_MSA.3 Static attribute initialisation

FMT_MSA.3 Static attribute initialisation

FMT_MSA.3 is iterated here to address the static attribute initialisation by the Secure Boot Loader software (FMT_MSA.3[SBL]).

FMT_MSA.3 [SBL]

Hierarchical to: No other components

FMT_MSA.3.1 The TSF shall enforce the *Control of Software download Policy* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the *none* to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 Management of security attributes
FM_SMR.1 Security roles

Application Notes: ‘None’ means that an initial value is not permitted to be changed.

FMT_MSA.1 Management of security attributes

FMT_MSA.1 is iterated here to address the management of security attribute by the Secure Boot Loader software (FMT_MSA.1 [SBL]).

FMT_MSA.1 [SBL]

Hierarchical to: No other components

FMT_MSA.1.1 The TSF shall enforce the *Control of Software download Policy* to restrict the ability to *modify* the security attributes *verification* to *the Secure Boot Loader software*.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1 is iterated here to address the specification of management functions by the Secure Boot Loader software (FMT_SMF.1 [SBL]).

FMT_SMF.1 [SBL]

Hierarchical to: No other components

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: *modification of the software download, by the Secure Boot Loader software, under the Control of Software download Policy*.

Dependencies: No dependencies

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 **ADV_FSP.5 Complete semi-formal functional specification with additional error information** ADV_IMP.1 Implementation representation of the TSF **ADV_INT.2 Well-structured internals** ADV_TDS.4 Semiformal modular design	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 **ALC_CMS.5 Development tools CM coverage** ALC_DEL.1 Delivery procedures ALC_DVS.2 Identification of security measures ALC_LCD.1 Developer defined life-cycle model **ALC_TAT.2 Compliance with implementation standards**	EAL5, PP EAL5 EAL5, PP PP EAL5 EAL5
ATE: Tests	ATE_COV.2 Analysis of coverage **ATE_DPT.3 Testing: modular design** 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, January 2013, Version 2.9.

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	– FDP_ITT.1 “Basic internal transfer protection”, – FPT_ITT.1 “Basic internal TSF data transfer protection”, – FDP_IFC.1 “Subset information flow control”
O.Phys-Probing	– FPT_PHP.3 “Resistance to physical attack”
O.Malfunction	– FRU_FLT.2 “Limited fault tolerance”, – FPT_FLS.1 “Failure with preservation of secure state”
O.Phys-Manipulation	– FPT_PHP.3 “Resistance to physical attack”
O.Leak-Forced	All requirements listed for O.Leak-Inherent – FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, plus those listed for O.Malfunction and O.Phys-Manipulation – FRU_FLT.2, FPT_FLS.1, FPT_PHP.3
O.Abuse-Func	– FMT_LIM.1 “Limited capabilities”, – FMT_LIM.2 “Limited availability”, plus those for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, – FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, FPT_PHP.3, FRU_FLT.2, FPT_FLS.1
O.Identification	– FAU_SAS.1 “Audit storage”
O.RND	– FCS_RNG.1 “Quality metric for random numbers”, plus those for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation, O.Leak-Forced – FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, FPT_PHP.3, FRU_FLT.2, FPT_FLS.1
O.Add-Functions	– FCS_COP.1 [3DES] “Cryptographic operation” – FCS_COP.1 [AES] “Cryptographic operation” – FCS_COP.1 [RSA Encryption and Decryption] “Cryptographic operation”
O.SWDetect	– FPT_FLS.1
O.SWResponse	– FPT_FLS.1

Objective	TOE Security Functional and Assurance Requirements
O.SWDownload	– FDP_ITC.1 – FDP_ACC.1 [SBL] – FDP_ACF.1 [SBL] – FMT_MSA.3 [SBL] – FMT_MSA.1 [SBL] – FMT_SMF.1 [SBL] – FCS_COP.1 [AES] "Cryptographic operation" – FCS_COP.1 [RSA signature verification] "Cryptographic operation"
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, AES and RSA requirement of O.Add-Functions is directly implemented by FCS_COP.1.1.

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.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.SWDetect is implemented by the exception conditions recognised under FPT_FLS.1 and the opportunities provided to test for violations of secure state (or to maintain velocity check parameters) by the EEPROM write interrupts in FPT_FLS.1. As noted for O.Phys-Manipulation and O.Malfunction above, the access control provisions in FDP_ACC.1 and FDP_ACF.1 provide protection for the ways in which these detection features are used. The restrictive default value in FMT_MSA.3 is provided by TOE. The security management in FMT_MSA.1 and FMT_SMF.1 provide to be configured by the Security IC Embedded Software.

O.SWResponse is implemented by the ability of Security IC Embedded Software to cause a reset (by setting the HLT 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.

O.SWDownload is implemented to download the Security IC Embedded Software under FPT_ITC.1. The access control provisions in FDP_ACC.1 and FDP_ACF.1 provide to permit the downloading of the Security IC Embedded Software into EEPROM. The restrictive default value in FMT_MSA.3 is provided by TOE. The RSA verification and AES decryption in FCS_COP.1 [RSA signature verification] and RCS_COP.1 [AES] provide integrity and confidentiality for the downloading of the Security IC Embedded Software. The security management in FMT_MSA.1 and FMT_SMF.1 provide to download the Security IC Embedded Software into EEPROM, only once in the product lifecycle.

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

SFR	Operation required	Operation performed
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	[selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic]	physical
	[selection: prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source, generates the internal random numbers with a post-processing algorithm of class DRG.2 as long as its internal state entropy guarantees the claimed output entropy]	prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source
	[selection: externally, at regular intervals, continuously, applied upon specified internal events]	applied upon specified internal events: start-up, before and after key generation
	[assignment: list of security capabilities]	PTG.2.1, PTG.2.2, PTG.2.3, PTG.2.4, PTG.2.5

SFR	Operation required	Operation performed
	[selection: bits, octets of bits, numbers [assignment: format of the numbers]]	numbers in the format 16-bit
	[assignment: a defined quality metric]	PTG.2.6, PTG.2.7
	[assignment: additional standard test suites]	as defined in [AIS31]
FCS_COP.1 [3DES]	[assignment: list of cryptographic operations]	Encryption and decryption
	[assignment: cryptographic algorithm]	Triple Data Encryption Standard (3DES) in one of the following modes of operation: ECB, CBC, OFB
	[assignment: cryptographic key sizes]	112 or 168 bit
	[assignment: list of standards]	Technology Administration U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm(TDEA) Block Cipher, Revised 19 May 2008, NIST Special Publication 800-67, Version 1.1 'DES MODES OF OPERATION' Federal Information Processing Standards Publication 81, 2nd December 1980 (http://www.itl.nist.gov/fipspubs/fip81.htm) National Institute of Standards and Technology (NIST), Technology Administration, U.S. Department of Data Encryption Standard (DES), NIST Special Publication 800-38A, 2001 Edition.
FCS_COP.1 [AES]	[assignment: list of cryptographic operations]	Encryption and decryption
	[assignment: cryptographic algorithm]	Advanced Encryption Standard (AES) in one of the following modes of operation: ECB, CBC, OFB
	[assignment: cryptographic key sizes]	128, 192, and 256 bits

SFR	Operation required	Operation performed
	[assignment: list of standards]	U.S. Department of Commerce / National Bureau of Standards Advanced Encryption Standard, FIPS PUB 197, 2001 November 26. National Institute of Standards and Technology Special Publication 800-38A 2001 Edition, Recommendation for Block Cipher Modes of Operation Methods and Techniques.
FCS_COP.1 [RSA Encryption and Decryption]	[assignment: list of cryptographic operations]	Encryption and decryption
	[assignment: cryptographic algorithm]	R.Rivest, A.Shamir and L.Adleman (RSA)
	[assignment: cryptographic key sizes]	from 1024 to 2176 bits
	[assignment: list of standards]	Encryption: Section 5.1.1 RSAEP in [RSA]. Decryption (without CRT): Section 5.1.2 RSADP in [RSA].
FCS_COP.1 [RSA signature verification]	[assignment: list of cryptographic operations]	Verification
	[assignment: cryptographic algorithm]	R.Rivest, A.Shamir and L.Adleman (RSA)
	[assignment: cryptographic key sizes]	2048 bits
	[assignment: list of standards]	Section 5.2.2. RSAVP1 in [RSA].
FDP_ACC.1 [SBL]	[assignment: access control SFP]	Control of Software download Policy
	[assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP].	the execution of Secure Boot Loader software instructions. Here, Policy: Control of software downloads Policy Subjects: Secure Boot Loader software Objects: the IC Embedded software Operations: to download

SFR	Operation required	Operation performed
FDP_ACF.1 [SBL]	[assignment: access control SFP]	Control of software download Policy
	[assignment: security attributes, named groups of security attributes]	Subject: Secure Boot Loader software Object: instruction attributes: execution
	[assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]	the Security Boot Loader Software can be executed downloading of the IC Embedded Software only, if the signature verification process is succeeded
	[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]	None
	[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]	None
FDP_ITC.1	[assignment: access control SFP(s) and/or information flow control SFP(s)]	Control of software download Policy
	[assignment: additional importation control rules]	none
FMT_MSA. 3[SBL]	[assignment: access control SFP(s), information flow control SFP(s)]	Control of software download policy
	[selection, choose one of : restrictive, permissive, [assignment: other property]]	restrictive
	[assignment: the authorised identified roles]	none
FMT_MSA. 1[SBL]	[assignment: access control SFP(s), information flow control SFP(s)]	Control of software download policy
	[selection: change_default, query, modify, delete, [assignment: other operations]]	modify
	[assignment: list of security attributes]	verification
	[assignment: the authorised identified roles]	the Secure Boot Loader software
FMT_SMF. 1[SBL]	[assignment: list of management functions to be provided by the TSF]	modification of the software download, by the Secure Boot Loader software, under the Control of Software download Policy

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)
FCS_COP.1 [AES]	[FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1] FCS_CKM.4	Yes (by the environment)
FCS_COP.1 [RSA Encryption and Decryption]	[FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1] FCS_CKM.4	Yes (by the environment)
FCS_COP.1 [RSA signature verification]	[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	Yes

SFR	Dependencies	Fulfilled by Security Requirements in this ST?
FDP_ITC.2	[FDP_ACC.1, or FDP_IFC.1] [FTP_ITC.1, or FTP_TRP.1] FPT_TDC.1.	No additional requirement – see discussion below
FCS_CKM.1	[FCS_CKM.2, or FCS_COP.1] FCS_CKM.4	No additional requirement – see discussion below
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2, or FCS_CKM.1]	No additional requirement – see discussion below
FDP_ACC.1 [SBL]	FDP_ACF.1	Yes
FDP_ACF.1 [SBL]	FDP_ACC.1 FMT_MSA.3	Yes
FMT_MSA.3 [SBL]	FMT_MSA.1 FMT_SMR.1	No additional requirement – see discussion below
FMT_MSA.1 [SBL]	[FDP_ACC.1, or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	No additional requirement – see discussion below
FMT_SMF.1 [SBL]	No dependencies	Yes

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.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.

The dependency FMT_SMR.1 introduced by the two components FMT_MSA.1 and FMT_MSA.3 is considered to be satisfied because the access control specified for the intended TOE is not role-based but enforced for each subject. Therefore, there is no need to identify roles in form of a security functional requirement FMT_SMR.1.

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.

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 or powered on. When an external 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]).

Application Notes:

- The maximum and minimum operating voltages are defined in [HM].
- The maximum and minimum operating frequencies are defined in [HM].
- The maximum and minimum operating temperatures are defined in [HM].
- "Failure against specification" means a failure to operate as specified in [HM].
- An illegal instruction is any operation code not defined in [SM].
- An illegal access is marked "–" in the table titled "Access" in the TOE memory map (section 4 of [HM]). In addition, among the accesses marked "✓" in the tables, if a type of access allowed is noted under the mark, an access other than the allowed type of access is also an illegal access.

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 device.
- CPU protection – software can instruct the TOE to insert additional measures in the CPU operation.

- DES protection – the DES coprocessor contains additional measures to resist side-channel attacks.[13]

- AES protection – the AES coprocessor contains additional measures to resist side-channel attacks.[14]

- RSA protection – the PKCC contains additional measures to resist Simple Power analysis and timing attacks.[15]

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 PTG.2 (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 11 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 TECB mode, TCBC mode and TOFB mode according to the following standard:

a) Technology Administration U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm(TDEA) Block Cipher, Revised 19 May 2008, NIST Special Publication 800-67, Version 1.1

b) 'DES MODES OF OPERATION' Federal Information Processing Standards Publication 81, 2nd December 1980 (http://www.itl.nist.gov/fipspubs/fip81.htm)

c) National Institute of Standards and Technology (NIST), Technology Administration, U.S. Department of Data Encryption Standard (DES), NIST Special Publication 800-38A, 2001 Edition.

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

[13] As with bus encryption, this protection is always active.
[14] As with bus encryption, this protection is always active.
[15] As with bus encryption, this protection is always active.

some application contexts, triple DES (as described in section 17.3.1 of [HM]) may be required in order to achieve a suitable strength of function.

2. To provide secure embedded software, the software developer is required to ensure that the triple DES, the AES and (where used in a cryptographic algorithm) PKCC, 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].

5. SF.AES

The TOE provides a hardware AES coprocessor that carries out AES encryption and decryption in ECB mode, CBC mode, and OFB mode according to the following standard:

a) U.S. Department of Commerce / National Bureau of Standards Advanced Encryption Standard, FIPS PUB 197, 2001 November 26.

b) National Institute of Standards and Technology Special Publication 800-38A 2001 Edition, Recommendation for Block Cipher Modes of Operation Methods and Techniques.

Application Notes

1. The TOE provides an AES coprocessor, it is only actually used as an encryption and decryption function in the context of particular Security IC Embedded Software.

2. See SF.DES Application Notes No.3.

6. SF.RSA

The TOE provides a hardware PKCC that carries out RSA encryption and decryption according to the following standard:

a) Section 5.1.1. RSAEP, section 5.1.2 RSADP in [RSA].

Application Notes

1. The TOE provides a PKCC, it is only actually used as an encryption and decryption function in the context of particular Security IC Embedded Software.

2. see SF.DES Application Notes No.3

3. The RSA function supports the encryption lengths from 192-bit to 2232-bit, but only the lengths from 1024-bits to 2176-bits are claimed as sufficiently secure, and at least 1976 bits are required for signature applications (see [SA]).

7. SF.ESFunctions

The scope of the TOE evaluation includes correct operation of aspects such as the CPU instructions, memory functions and standard peripherals such as memories, registers, I/O interfaces, timers and UART as specified in [HM]. In addition, the TOE offers hardware and software facilities that are designed to enable 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 HLT bit of the System Control Register 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 20.5.2 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.

8. SF.TestModeControl

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

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

9. 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.

10. SF.SBL

    The TOE provides the Secure Boot Loader software to ensure the control of the software download into the EEPROM memory. The software download process can process only after verification process has succeeded. The Secure Boot Loader software ensures the integrity of user by RSA verification. The Secure Boot Loader software ensures the integrity of user data by AES encryption.

    The Secure Boot Loader is based on the usage of a secret key (AES) for encryption of the customer code and an asymmetric key pair (RSA) used for signature verification. The Secure Boot Loader software uses the hardware PKCC and AES coprocessor that carries out RSA verification and AES encryption according to the following standard:

    For RSA verification by using PKCC

a) Section 5.2.2. RSAVP1 in [RSA].

For AES encryption by using AES coprocessor

b) U.S. Department of Commerce / National Bureau of Standards Advanced Encryption Standard, FIPS PUB 197, 2001 November 26.

c) National Institute of Standards and Technology Special Publication 800-38A 2001 Edition, Recommendation for Block Cipher Modes of Operation Methods and Techniques.

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, FCS_RNG.1
SF.LeakProtect	FDP_ITT.1, FPT_ITT.1, FDP_IFC.1
SF.RNG	FCS_RNG.1
SF.DES	FCS_COP.1 [3DES]
SF.AES	FCS_COP.1 [AES]
SF.RSA	FCS_COP.1 [RSA Encryption and Decryption]
SF.ESFunctions,	FRU_FLT.2, FPT_FLS.1, FPT_PHP.3
SF.TestModeControl	FMT_LIM.1, FMT_LIM.2
SF.Inject	FAU_SAS.1
SF.SBL	FCS_COP.1[AES], FCS_COP.1[RSA signature verification], FDP_ITC.1, FDP_ACC.1[SBL], FDP_ACF.1[SBL], FMT_MSA.1[SBL], FMT_MSA.3[SBL], FMT_SMF.1[SBL]

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.HWProtect, SF.RNG
FCS_COP.1 [3DES]	SF.DES
FCS_COP.1 [AES]	SF.AES, SF.SBL
FCS_COP.1 [RSA Encryption and Decryption]	SF.RSA
FCS_COP.1 [RSA signature verification]	SF.SBL
FDP_ACC.1 [SBL]	SF.SBL
FDP_ACF.1 [SBL]	SF.SBL
FDP_ITC.1	SF.SBL
FMT_MSA.3 [SBL]	SF.SBL
FMT_MSA.1 [SBL]	SF.SBL
FMT_SMF.1 [SBL]	SF.SBL

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] A proposal for: Functionality classes for random number generators, Version 2.0, Bundesamt für Sicherheit in der Informationstechnik (BSI), 18 September, 2011

[CC/1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, September 2012, version3.1, revision4, CCMB 2012-09-001

[CC/2] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, September 2012, version3.1, revision4, CCMB 2012-09-002

[CC/3] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, September 2012, version3.1, revision4, CCMB 2012-09-003

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

[RSA] PKCS #1 v2.1: RSA Cryptography Standard, RSA Laboratories, 14 June, 2002

[SA] Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen, 6th January 2010

Document

[HM] RS4FC128, RS4FC128E User’s Manual: Hardware, Rev.1.00, July, 2013

[OPT] Option List for Smart Card Microcomputer (for RS4FC128), Rev.0.2, Renesas Electronics Corporation, 16 November, 2012

[SBLM] Secure Boot Loader Version 5560 User’s Manual: Renesas Secure Microcomputer RS-4E Series, Rev. 1.10, Renesas Electronics Corporation, 8 August, 2013

[UGM] RS-4E Series User Guidance Manual, Rev. 1.1, Renesas Electronics Corporation, 19 September, 2013

[SM] H8S/2600 Series H8S/2000 Series Software Manual Rev.4.00, Renesas Technology Corp., 24 February, 2006

8.2 Others

None.

** End of Document **