debis

Services by DaimlerChrysler

Certification Report

Setsos 2.1 Security Module Operating System

Setec Oy

debisZERT-DSZ-ITSEC-04010-1998

debis IT Security Services

The Modern Service Provider

Preface

The product *Setsos 2.1 Security Module Operating System* of *Setec Oy* has been evaluated against the ITSEC. The evaluation has been performed under the terms of the certification scheme debisZERT of debis IT Security Services. The certification procedure applied conforms to the rules of service type 4: Certificates recognized by the Bundesamt für Sicherheit in der Informationstechnik (BSI).

The result is:

Security Functionality:	Access Control
Assurance Level:	E3
Strength of Mechanisms:	Type B mechanisms: impregnable to direct attack if perfectly conceived and implemented

For further information and copies of this report, please contact the certification body:

✉ debis IT Security Services
- Certification Body -
Rabinstr. 8
D-53111 Bonn
Germany

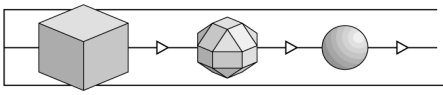
✆ +49-228-9841-110
Fax: +49-228-9841-60
Email: debiszert@itsec-debis.de
WWW: www.itsec-debis.de

This is to certify that the evaluation has been performed compliant to the scheme debisZERT.

Bonn, 22 October 1998

Certifier: Klaus-Werner Schröder

Head of the
Certification Body: Dr. Heinrich Kersten



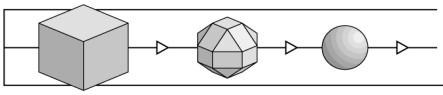
Revision List

The following revision list shows the history of this certification report.

Information on re-certifications due to product modifications are given in chapter 7.

Revision	Date	Activity
0.9	23.09.98	Preversion (based on template report 1.3)
1.0	24.09.98	Initial release (based on template report 1.3)
1.1	21.10.98	Proof corrections (based on template report 1.3)
1.2	22.10.98	Final update (based on template report 1.3)

Contents



(This page is intentionally left blank.)

1 Introduction

1.1 Evaluation

1 The evaluation was sponsored by Setec Oy, P. O. Box 31, FIN-01741 Vantaa, Finland.

2 The evaluation was carried out by the evaluation facility debis Systemhaus Information Security Services GmbH, Prüfstelle für IT Sicherheit, and completed on September 14, 1998.

3 The evaluation has been performed against the ITSEC and ITSEM. Some explanations concerning the contents of ITSEC and ITSEM can be found in chapter 5.

1.2 Certification

4 The certification was performed under the terms of the certification scheme debisZERT of debis IT Security Services. The Certification Body of debis IT Security Services complies to EN 45011 and was accredited with respect to this standard by the Deutsche Akkreditierungsstelle für Informations- und Telekommunikationstechnik (DEKITZ) under DAR Registration Number DIT-ZE-005/98-00.

5 The Certification Body applied the certification procedure as specified in the following documents:

/Z01/ Certification Scheme

/V04/ Certificates recognised by the BSI

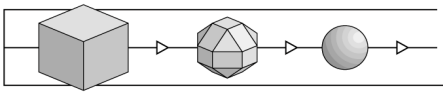
1.3 Certification Report

6 The certification report states the outcome of the evaluation of Setsos 2.1 Security Module Operating System - referenced as TOE = Target of Evaluation in this report.

7 The certification report is only valid for the specified version of the TOE. It can be extended to new or different versions as soon as a successful re-evaluation has been performed.

8 The certification report is intended

- as a formal confirmation for the sponsor concerning the performed evaluation,
- to assist the user of Setsos 2.1 Security Module Operating System when establishing an adequate security level.



9 The certification report contains pages 1 to 44. Copies of the certification report can be obtained from the sponsor or the Certification Body.

10 The certification report can be supplemented by statements of successful re-certification and by annexes on special technical problems. Such statements and annexes will be published in

/Z02/ Certified IT Products, Systems and Services.

1.4 Certificate

11 A survey on the outcome of the evaluation is given by the security certificate debisZERT-DSZ-ITSEC-04010-1998.

12 The contents of the certificate are published in the document

/Z02/ Certified IT Products, Systems and Services

and on the WWW.

13 The certificate is formally recognised by the Bundesamt für Sicherheit in der Informationstechnik (BSI) that confirms the equivalence of this certificate to its own certificates in the international context.

14 The certificate carries the logo officially authorised by the BSI. The fact of certification will be listed in the brochure BSI 7148.

1.5 Application of Results

15 The processes of evaluation and certification are performed with current expertise, but cannot give an absolute guarantee that the certified object is free of vulnerabilities. With increasing evaluation level the probability of undiscovered exploitable vulnerabilities decreases.

16 It is highly recommended to read the certification report carefully to benefit from the evaluation. In particular, the information provided on the intended method of use, the assumed threats, the operational environment and the evaluated configurations are essential for the user.

17 The results of the evaluation are only valid under the assumption that all requirements specified in the certification report are met by the user.

Otherwise, the results of the evaluation are not fully applicable. In such a case, there is a need of an additional analysis whether and to which degree the certified object can offer security under the modified assumptions. The evaluation facility and the Certification Body can give support to perform this analysis.

2 Evaluation Findings

2.1 Introduction

18 The outcome of the evaluation is represented by the ETR (Evaluation Technical Report). The evaluation was performed against the security target specified in chapter 3.

2.2 Evaluation Results

19 The evaluation facility came to the following conclusion:

- The TOE meets the requirements of the assurance level **E3** according to ITSEC, i.e. all requirements at this assurance level as to correctness and effectiveness are met:

  ITSEC E3.1 to E3.37 for the correctness phases

  Construction - The Development Process (Requirements, Architectural Design, Detailed Design, Implementation),

  Construction - The Development Environment (Configuration Control, Programming Languages and Compiler, Developers Security),

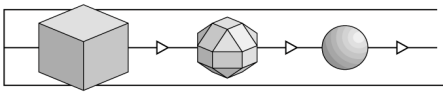
  Operation - The Operational Documentation (User Documentation, Administration Documentation)

  Operation - The Operational Environment (Delivery and Configuration, Start-up and Operation).

  ITSEC 3.12 to 3.37 for the effectiveness with the aspects

  Effectiveness Criteria - Construction (Suitability of Functionality, Binding of Functionality, Strength of Mechanism, Construction Vulnerability Assessment),

  Effectiveness Criteria - Operation (Ease of Use, Operational Vulnerability Assessment).



- The mechanisms of the TOE M1, M3, and M3a are critical mechanisms; they are of type B. The remaining mechanisms M2 and M4 are non-critical.

  For mechanisms of type B no rating of strength is specified in accordance with ITSEM. But even if an attack potential according to level “medium” was considered in the vulnerability analysis phase, no exploitable vulnerability was detected in the assumed environment (cf. chapter 3, Security Target).

2.3 Further Remarks

20 The evaluation facility has formulated the following requirement to the sponsor: The SAM products should not be delivered to a customer without the TOE being implemented in the ROM of a smart card.

21 The evaluation facility has formulated no further requirements to the user.

3 Security Target

3.1 Introduction

This document is an ITSEC security target of Setsos 2.1 security module operating system.

3.1.1 References

[1] "M68HC05SC family, Technical Summary, 8-bit microcontroller family with security features", Motorola Ltd., 1994.
[2] "Smartcards and security", Motorola Ltd.
[3] "Security requirements for products with the algorithm SLE443X or compatible", version 1.0, Siemens AG, 9.2.1995.

3.1.2 Definitions

SAM

Secure Application Module. A term "security module" is also used to refer to SAM. The SAM is a smart card component which is located inside a terminal and which operates as a counterpart of a user carried smart card or memory card.

The SAM contains the operating system (TOE) and a file system. The operating system is a fixed built-in part of the SAM. The file system is set up in initialization of the SAM, before usage. The file system is hierarchical. Subdirectories of the root directory are called applications, because they form independent data and program entities. This property is the foundation of the multi-application aspect of the SAM.

The SAM contains application programs which can use the internal files of the application. The files can contain e.g.

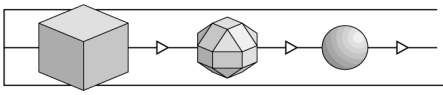
a) DES keys, for example master keys for Eurochip card key computation,
b) counter values,
c) log records and
d) other freely formatted data.

The programs can use also resources outside the SAM via the command interface.

SAPL

SAM Application Programming Language. A proprietary programming language that is used to define programs to be interpreted and executed by a Setec's SAM. The SAPL instruction set is defined in the operational documentation.

The SAPL programs are



a) written by a text editor,
b) converted to object format by a compiler program,
c) stored in a program file of the SAM and
d) interpreted (executed) internally by the SAM.

Execution of the program is initiated by the terminal via the command interface.

The SAPL provides a method to construct executable application programs which

a) are compact,
b) cannot compromise the confidentiality of other applications or the SAM itself and
c) cannot compromise the integrity of other applications or the SAM itself.

The execution of SAPL programs in a SAM could be considered equal to running executable programs in a microcontroller with the following analogy:

a) The SAPL programs equals the executable, defining which instructions are executed in which order to produce the desired functionality.
b) The Setsos 2.1 operating system (TOE) equals the microcode of the microprocessor, defining the effect of each instruction and the resources to be accessed by the instructions.

program interpreter

The SAPL programs are executed by the built-in program interpreter in the TOE. It reads one instruction code at a time from the program file and calls the corresponding routine in the TOE. In principle there is one routine for each recognized instruction code, but in practice some instructions can share one routine when the differences in operation are minimal.

The execution of SAPL programs is related with

a) set of accessible internal data files,
b) internal working memory and
c) global internal variables, of which the program counter is one.

command interface, serial interface

Term used to denote

a) the I/O link between the SAM and the terminal,
b) the protocol used for communication over a) and
c) the set of defined commands that are transmitted using a) and b).

The command interface serves both for

a) the terminal to access the SAM,
b) the SAM programs (SAPL) to access the external resources.

The characteristics of the I/O line and the protocol are defined in ISO standard ISO/IEC 7816-3.

program execution interface

Term used to denote

a) the method of executing SAPL programs and
b) the set of instructions that are defined for SAPL.

In other words, the SAPL programs or any part of them can be executed only by the SAM internally, but the initiation to do so is originated from the terminal. Conversely, the actions that are related to the program execution interface can be performed only by starting a SAPL program and having the corresponding instruction in the SAPL program.

terminal

Term used to denote the hardware to establish an electrical connection to the SAM. The terminal is not any specific device, but can be viewed as the media to access the SAM. Because the text in this document views many things from the point of view of the SAM, the terminal represents any external entity.

For example, when “terminal initiates execution of SAPL program via the command interface”, the terminal is merely the media between the actual initiator (human, computer process) and SAM, but it is the device that eventually sends the commands to the SAM via the command interface.

ICC

Integrated Circuit Card. Equals the term smart card.

Eurochip, Eurochip card

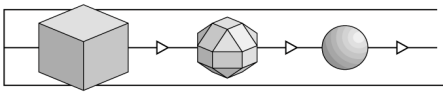
The Eurochip algorithm is a one-way authentication algorithm which is incorporated in

a) payment system security modules and
b) memory cards.

The cards are disposable cards which do not contain a microprocessor like real smart cards. The algorithm provides a method to distinguish authentic cards from replicas.

The algorithm produces a response from

a) challenge information,
b) static card information,



c) secret (static) card key and
d) card balance information,

thus providing a dynamic authentication.

The algorithm is contained in the TOE for the purpose of verifying the correctness of a Eurochip card authentication. As the algorithm is highly confidential, any devices containing an instance of it must be evaluated. The security objectives and threats of this security target are dictated by the requirements of such an evaluation, defined by Siemens (see [3] p. 4).

SAM application

The SAM has a hierarchical file structure, where the Master File (MF) is the root directory, and it has Elementary Files (EF) and Dedicated Files (DF) as its files. The DFs are directories that can contain only EFs, which are files that store the keys, SAPL programs, balance information, SAM identification and other application related data. An application refers to one DF and its files. The file system is compliant with the ISO standard 7816-4 and CEN standard 726-3.

A SAPL program in a program file of a DF can access only the files of the DF, not files of MF or any other DF. Thus a program can access only files of the application it is part of.

master key

A master key is the key from which the keys of all cards are derived. See “card key”.

card key

All keys in the cards are diversified keys, meaning that they are derived from master keys. The card keys are computed by running a cryptographic algorithm for the master key and card identification information. In this way the entity in possession of the master key can compute the key in every card when given the card identification information.

pre-initialization phase

The SAM containing the TOE has a cleared EEPROM memory after manufacture. Before the initialization the EEPROM can be tested by filling it with a pattern and comparing the contents against a pattern. This is to allow detection of weak components before initialization phase.

In the pre-initialization phase

a) the EEPROM bytes can be compared against a bit pattern,
b) the EEPROM can be filled with a bit pattern and

c) the EEPROM write-erase cycles can be performed to stress the bytes.

The pre-initialization phase is the initial phase. It is exited when the transport code is sent to the SAM, entering the initialization phase. The transition from pre-initialization to initialization phase forces the erasure of the EEPROM, except the first 4-byte block which contains the counter for transport code presentation attempts. Abortion of initialization sequence in initialization phase returns the SAM back to pre-initialization phase. Also the ERASE_SAM command of normal usage phase can be used to return the SAM back to pre-initialization phase, which also clears the transport code attempts counter.

initialization

The SAM containing the TOE must be prepared for normal usage phase. In the initialization

a) the file system (including master keys for Eurochip card key computation),
b) the parameters regulating the EEPROM programming time,
c) the secret code controlling the total erasure of EEPROM (for reuse),
d) the random number generator seed values (for random challenge generation during normal usage phase),
e) the answer-to-reset (ATR) bytes for normal usage phase and
f) the flag indicating the normal usage phase (to exit initialization phase)

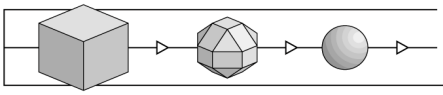
are set up via the command interface. The initialization affects only the contents of the EEPROM memory of the SAM.

The random number generation is used, among other things, during the memory card authentication, and is security relevant in context of the memory card authentication as a whole, but not in the context of this evaluation.

The normal usage indicating flag in the EEPROM forces the SAM to accept only normal usage commands after restart, thus entering normal usage phase. This flag can be written only during initialization phase and erased only together with all other contents of the EEPROM in the normal usage phase by the command ERASE_SAM. Setting or clearing this flag has no effect on the security objectives of this evaluation.

For authorisation of initializer a transport code factor is programmed in the SAM by the integrated circuit (IC) component manufacturer according to supplier's (Setec Oy) instructions. Attempts to send the transport code are recorded by the SAM, allowing a maximum of 8 false attempts before irrecoverable lock-up. These features prevent initialization by other parties than Setec Oy. The initialization consists of

a) presenting a transport code to the SAM via the command interface,
b) issuing several EEPROM writing commands via the command interface to initialize the EEPROM contents, and
c) computation of internal checksums (by TOE) and starting the normal operation of SAM by a command via the serial interface.



The step c) can be replaced by resetting of the SAM if the initialized data already contains the correct checksum values (computed by the initialization system). The internal checksums provide additional protection of the integrity of the file system and aim to prevent the TOE from using corrupted information. The checksums are non-linear 8-bit hash values, computed from some hundreds of bytes of EEPROM memory. These checksums are of no relevance to the security objectives.

The Eurochip algorithm cannot be accessed before or during the initialization via the serial interface, because the TOE does not contain any function to access the algorithm with pre-initialization or initialization phase commands. After the initialization it can be executed, but not read or modified via the serial interface.

3.2 Product rationale

3.2.1 The target of evaluation

The target of evaluation (TOE) is the Setsos 2.1 security module operating system.

The TOE is realized in the ROM memory of a physically secure microcontroller.

The TOE is a smart card like operating system, which incorporates both

a) external command handling and
b) internal handling of special instruction code of a command file.

This instruction code format is called the SAM Application Programming Language (SAPL). Specification of this language is contained in the operational documentation.

For SAPL execution there is a library of routines in ROM memory of SAM, one for each SAPL instruction, which are called when reading the corresponding instruction from the program file. These routines include manipulation of data in

a) the files of the SAM application,
b) the RAM memory of the SAM and
c) the files of an external smart card.

Smart card manipulating routines include sending commands from SAM via the serial interface.

The SAPL realizes a programming language to manipulate the files of the SAM application and files of a smart card in a well defined way, without allowing execution of machine code from a file, thus avoiding introduction of unauthorized operations in application programs.

The TOE includes the following security features:

The TOE provides two modes of access:

a) defined commands given via the serial interface, of which some are usable prior to normal usage and some during normal usage, and

b) a built-in interpreter mechanism for program file execution.

The program execution is initiated by one of the normal usage phase commands.

All of the operating system is contained in the hardwired mask ROM of the chip. It is therefore impossible to modify the operation of the TOE, including the algorithm, via the serial interface.

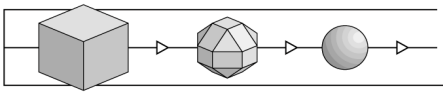
The TOE does not provide any channel to read out or modify the algorithm. The command interface or the program interpreter do not contain any method for those operations.

The TOE implements only such execution interface to the algorithm, that it can be used for verification of a readily performed memory card authentication only. Therefore the use of algorithm is limited in such a way that the TOE is giving authentication verification result as logical yes/no indication. No more than this binary value can be exported from the TOE in all cases of correct or even manipulated execution. All internal calculation data is destroyed during the authentication verification and also in start-up of the TOE.

The delivered TOE consists of software and documentation listed in the following table[1].

type	name	submitted as	date	identification
SW	Setsos 2.1	integrated into a smart card	09.10.1996	ROM release 96-10-09
doc	Command interface, Setsos 2.1 security module operating system	manual, file	23.10.1997	Version 1.0
doc	Technical description, Setsos 2.1 security module operating system	manual, file	3.12.1997	Version 1.0
doc	SAPL manual, Setsos 2.1 security module operating system	manual, file	28.11.1997	Version 1.0
doc	Customer documentation information, Setsos 2.1 security module operating system	manual, file	24.03.1998	Version 1.1

1 The certification body updated the table according to the certification process.



3.2.2 Intended method of use

The TOE is to be used as the operating system of a payment system security module. It is meant to operate as a secure counterpart of a smart card or a Eurochip memory card to provide an interface to the payment system for the holders of authentic cards.

The TOE can perform an authenticity check of a memory card incorporating an Eurochip algorithm SLE 4436. When given

a) the master key,
b) the card data,
c) the challenge given to the card and
d) the response of the card

it can verify the validity of the response. The TOE gets a master key from an internal file and computes the card key internally, provided that the card belongs to the same system sharing the master key. The verification produces only a binary result (correct / incorrect). The verification is accessed via the SAPL program execution interface.

There are three major phases in the life cycle of the SAMs:

a) pre-initialization phase,
b) initialization phase and
c) normal usage phase.

In the pre-initialization phase the EEPROM of the SAM can be stressed and tested. In the initialization phase the SAM is prepared for the normal usage phase (see also chapter 3.1.2). For reuse it is possible to restore the SAM to the initial state, thus re-entering the pre-initialization phase. The phases, actions performed in the phases or transitions between the phases do not have any effect on the TOE satisfying the security objectives.

No access to the assets - the Eurochip algorithm - except execution as a verification routine, is provided by the TOE. Therefore no access rights via access tables need to be defined in any phase regarding the algorithm. Transitions from pre-initialization to initialization and from normal usage to pre-initialization phase are related with secret codes, but these are related to the phase transitions only.

The initialization must be performed prior to using the SAM in the payment system by the SAM supplier (Setec Oy). After that the SAM can be installed in a terminal device by the terminal manufacturer under control of the payment system operator.

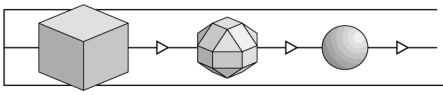
In the normal usage phase the terminal issues commands to the SAM via the command interface to execute the application programs. The SAM can contain application programs for various purposes. The Eurochip authentication can be incorporated in any of the programs, but it is intended to be used in context of purchasing event using a Eurochip memory card. In this case the sequence of actions is in practice the following (note that this refers to payment system details and is security irrelevant for this evaluation):

a) the payment device is started up by the merchant or the administrator of an automatic payment device,
b) the terminal starts up the SAM (and the TOE),
c) the card holder inserts a Eurochip card in the terminal,
d) the merchant or the (automatic) payment device enters the sum to be charged,
e) the terminal initiates the execution of the program, giving the sum to be charged from the card,
f) the SAM executes the program instructions, including among others
   1: checking card authenticity and balance:
      – reading of the card information,
      – deduction of the card balance,
      – checking that card balance is greater than or equal to the sum to be charged,
      – sending authentication challenge to the card,
      – reading authentication response from the card,
      – performing the verification of authentication with the Eurochip algorithm;
   2: decreasing card balance:
      – decreasing the balance of the card by the sum to be charged;
   3: checking the correct execution of the balance decreasing:
      – reading of the card information,
      – deduction of the card balance,
      – sending authentication challenge to the card,
      – reading authentication response from the card,
      – performing the verification of authentication with the Eurochip algorithm,
      – checking that the difference of the old and new balance of the card equals the charged sum;
   4: increasing the balance of the merchant:
      – increasing the balance of the SAM (decreased amounts are normally stored cumulatively in SAM);
g) the SAM returns information of the charging to the terminal,
h) the merchant or the (automatic) payment device grants the goods or services to the card holder.

Eventually, when reuse of the SAM components is needed, or when the SAM components are destroyed and deletion of stored information is needed, the SAM can be delivered back to the supplier who can erase the EEPROM contents totally.

3.2.3 Intended operational environment

The TOE is intended to be contained in the ROM memory of a smart card component, forming a secure application module (SAM). The integrated circuit (IC) component to be used is the Motorola MC68HC05SC28 (see [1] p. 1-9, 14). It has a single chip structure and several security features to prevent disclosure of the memory area contents.



The TOE provides only logical protection. The smart card component in which the TOE is realized provides the physical protection. The threat of direct physical attack cannot be estimated actually in the context of this security target, but it is very low due to the sophisticated security features of the component (see [1] p. 1, 2, 8 and [2] p. 1, 2).

There are two supporting technical security measures of the IC component that are used:

a) The entire TOE is implemented in ROM of the IC, thus preventing any changes via the serial interface after manufacturing of the IC. As the operating system never executes code from RAM or EEPROM, adding extra functionality in the IC during or after initialization that would violate the security objects is impossible.

b) The so-called address lock-out mask option of the component is used, which prevents all read and write accesses to ROM and EEPROM areas by an executable code running in the RAM or EEPROM. This is realized with flags in such part of the EEPROM area which is read-only memory (ROM). These flags have been set by the IC manufacturer and cannot be modified by application program or the TOE.

These measures are overlapping in all other respects, but a) also prevents using the algorithm for purposes other than that defined in the TOE.

Only the physical protection provided by the IC component and the supporting technical security measures a) and b) above are necessary for the TOE to satisfy the security objectives of this security target. The remaining part of this chapter describes measures and requirements that are relevant only for the availability of TOE services and security of the payment system in which the TOE is used.

The terminal which interacts with the SAM must follow the ISO/IEC 7816-3 standard in order to establish a connection with the TOE. It must also be able to apply the commands specified in the architectural design and operational documentation of the TOE in order to run the application programs of SAM. Furthermore it should be able to accept and manipulate the commands from SAM to other components to support the capabilities of the TOE to maximum extent. Also these commands have been specified in the architectural design and operational documentation of the TOE.

In order to perform successful verification of memory card authentication the memory card must contain a key derived from the master key that has been stored in the SAM file system (in cleartext, not encrypted; see statements in the initialization context later in this chapter).

The SAM incorporating the TOE must be prepared for normal usage by supporting procedural measures. The preparation phase is called the initialization of the SAM and is the responsibility of the SAM supplier (Setec Oy). The normal usage phase is entered when the data in EEPROM represents the initialized state of the SAM (see also chapter 3.2.2 and definition of initialization in chapter 3.1.2).

For reuse the SAM EEPROM can be totally cleared via the command interface by issuing a secret erasing code that is defined during initialization. After this operation the SAM irreversibly enters an uninitialized state (the pre-initialization phase) and forces all old data to be erased.

The applications that are initialized in the SAM must contain proper file structures and SAPL programs for correct operation. Note that any invalid data cannot compromise the security objectives of this security target, but desired functionality may not be achieved.

As the initialization includes writing the master keys for authentication of cards,

a) the initialization must be performed in a physically secure environment,
b) the sensitive data must be handled and stored securely outside the SAM to avoid disclosure and
c) the sensitive data must be transmitted in an encrypted format via the serial interface to the SAM, in which they are then stored in non-encrypted format.

Note that revelation of the master keys does not compromise the security objectives of this security target, but it threatens the security of the payment system using the SAMs.

When used in a payment system, any sensitive data must not be sent without encryption over the serial interface of the SAM. The sensitive data includes among other things

a) the secret keys and
b) the user entered card PIN numbers.

Note that sending such data in cleartext cannot compromise the security objectives of this security target, however.

3.2.4 Subjects, objects and access modes

The **subjects** are

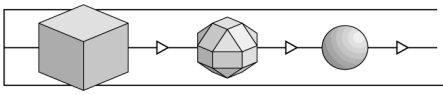
S1 the TOE process,
S2 the SAM pre-initializing and initializing terminal process, and
S3 the terminal process related to user and card.

Additional hypothetical subjects are

HS5 the chip manufacturer and
HS6 the direct probing process.

The **objects** are

O1 the ROM contents, including the algorithm,
O2 the RAM contents,
O3 the EEPROM contents,



O4 the pre-initialization and initialization commands, responses and EEPROM contents, and
O5 the normal usage commands, responses and data.

The **access modes** are the following:

A1 *reading* memory contents,
A2 *executing* memory contents,
A3 *writing* memory contents,
A4 *sending* commands, and
A5 *receiving* commands

Additional hypothetical access modes are

HA6 implanting (*writing*) the ROM mask and
HA7 direct probing (*reading*) of the ROM mask.

	O1	**O2**	**O3**	**O4**	**O5**
S1	A1, A2, A3[2]	A1, A2, A3	A1, A2[3], A3	A5	A4, A5
S2				A4	
S3					A4, A5
HS5	HA6				
HS6	HA7				

Table 1. Subjects, objects and respective access modes.

The table 1 above and the figure 1 below define the access modes that are available for the subjects for each object. The subjects HS5 and HS6, and thus access modes HA6 and HA7, are hypothetical and out of scope of this evaluation, because they are related to the manufacture and physical protection of the IC component in which the TOE is realized. The mentioned subject and access mode are included for completeness and as aid for understanding the entire relationship between the TOE and the externals.

Note that term 'subjects' does not refer to these hypothetical subjects, and the term 'access modes' does not refer to these hypothetical access modes in the following chapters.

3.2.5 Threats

The evaluation concerns the realization and accessibility of the Eurochip algorithm. The threats have been predefined by Siemens AG, and they are [4]:

---

2 The TOE process (S1) could in principle also execute (A3) writing instructions on the ROM area (O1). Such instructions would not have any effect on the contents of the ROM, however.

3 The access mode A2 for S1 handling O3 is not applicable in reality because of the lock-out mechanism mentioned in chapter 3.2.3.

T1 readout of the algorithm,
T2 alteration of the algorithm, and
T3 use of the algorithm for purposes other than verification of authenticity.

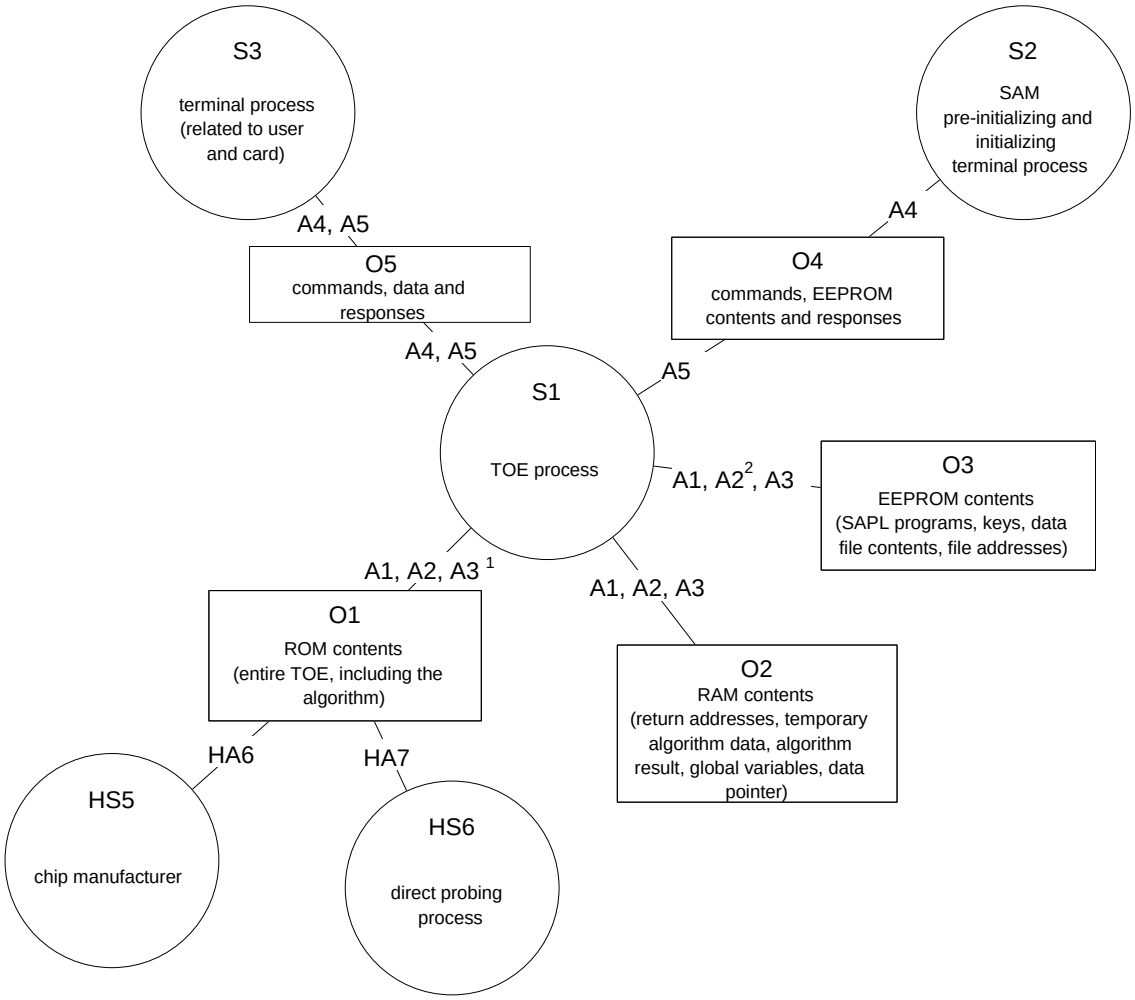
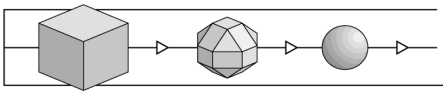


Figure 1. Subjects, objects and access modes, including the hypothetical ones (see footnotes 2 and 3 on page 15).

The *threat T1* violates the confidentiality requirement of the algorithm. Readout is defined as outputting the algorithm machine code or part of it via the serial interface. T1 is related to

a) the SAM pre-initialization and initializing terminal process (S2) sending (A4) pre-initialization and initialization commands (O4) to initialize an illegal file address causing b),
b) the TOE process (S1) reading (A1) the ROM memory (O1) at illegal address under control of information obtained by the TOE (S1) reading (A1) the EEPROM memory (O3),

4 Here is meant only the threats concerning the TOE, irrespective of possibility to, say, read out the algorithm from any other source, for example a Eurochip memory card that may be used in the payment system.



c) the TOE process (S1) reading (A1) the ROM memory (O1) at algorithm code address range as result of execution (A2) of ROM contents (O1) at illegal address,
d) the TOE process (S1) reading (A1) the ROM memory (O1) at algorithm code address range as result of execution (A2) of RAM contents (O2),
e) the TOE process (S1) reading (A1) the ROM memory (O1) at algorithm code address range as result of execution (A2) of EEPROM contents (O3),
f) the TOE process (S1) sending (A4) and the terminal process (S3) receiving (A5) normal usage commands (O5) containing part of ROM (O1) as consequence of b), c), d) or e),
g) the terminal process (S3) sending (A4) and the TOE process (S1) receiving (A5) normal usage commands (O5) to obtain part of ROM (O1) as consequence of b), c), d) or e).

The *threat T2* violates the integrity, but indirectly also the confidentiality requirement by modifying the algorithm so that its properties could be deduced from the output after modification. Alteration is defined as modification of the algorithm machine code via the serial interface. It is related to

a) the SAM initializing process (S2) sending (A4) pre-initialization and initialization commands (O4) to initialize an illegal file address causing b),
b) the TOE process (S1) writing (A3) to illegal write address in the ROM memory (O1) under control of information obtained by the TOE (S1) reading (A1) the EEPROM memory (O3),
c) the TOE process (S1) writing (A3) the ROM memory (O1) as result of execution (A2) of ROM contents (O1) at illegal address,
d) the TOE process (S1) writing (A3) the ROM memory (O1) as result of execution (A2) of RAM contents (O2),
e) the TOE process (S1) writing (A3) the ROM memory (O1) as result of execution (A2) of EEPROM contents (O3),
f) the terminal process (S3) sending (A4) normal usage commands (O5) modifying a part of TOE (O1) as consequence of b), c), d) or e),
g) the SAM initializing process (S2) storing to ROM address causing the TOE process (S1) to write (A3) in ROM memory (O1).

The interpretation of the *threat T3* is the usage of the algorithm or part of it for purpose of producing correct authentication responses of a given card related to a given authentication challenge. This means that the SAM could be used as a part of a counterfeiting device simulating an authentic card.

The threat T3 is related to

a) the TOE (S1) executing (A2) the algorithm in ROM (O1) only partly and
b) the TOE (S1) reading (A1) RAM memory (O2) after execution of the algorithm, resulting in reading temporary algorithm execution data.

3.2.6 Security objectives

The security objectives have been predefined by Siemens AG, and they are

SO1 protection of the algorithm against being read out via the interfaces of the TOE must be ensured (protection of confidentiality),
SO2 protection of the algorithm against alteration out via the interfaces of the TOE must be ensured (protection of integrity), and
SO3 it must be ensured that the algorithm is utilized for purposes of verifying authenticity only (protection against misuse and protection of confidentiality).

3.3 Security enforcing functions

The security enforcing functions (SEFs) of the TOE are all related to the generic heading "access control". The SEFs F1 to F3 are described below.

In the descriptions of the SEFs below, the term "illegal address" stored in EEPROM is an address that accesses memory areas other than the EEPROM. Code of TOE that has been intended for execution by the developer are referred to as "legal part" of TOE. The "illegal part" of TOE refers to execution of code that is achieved by misaligned or manipulated code address (executing an operand of an operation as instruction, possibly also using instructions as operands, or execution of data table contents).

F1 *Inhibition of read access to algorithm*

The TOE recognizes and denies read access to the algorithm code when

a) illegal addresses stored in EEPROM memory are used or
b) any legal or illegal part of TOE code is executed.

This function implements unconditional access control and prevents reading for all subjects.

(Note: There is no other possibility to read out the algorithm code via the serial interface.)

F2 *Inhibition of write access to algorithm*

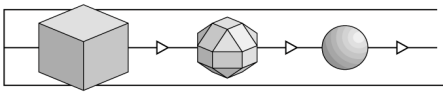
The TOE recognizes and denies write access to the algorithm code when

a) illegal addresses stored in EEPROM memory are used or
b) any legal or illegal part of TOE code is executed.

This function implements unconditional access control and prevents writing to algorithm for all subjects.

(Note: There is no other possibility to modify the algorithm code via the serial interface.)

(Note: The first of the two supporting technical security measures of chapter 3.2.3 effectively implements the same functionality.)



F3 *Inhibition of using the algorithm for other than verification purpose.*

The algorithm code execution

a) includes the verification of the result internally,
b) prevents examination of algorithm output, except the status of a), and
c) returns the TOE in its initial state unless the TOE execution is continued correctly.

Therefore the algorithm can be executed only so that it performs a verification and no other calculation data than status of the verification is revealed.

This is unconditional access control functionality. Execution of the algorithm for verification purpose is free and execution for other purposes is prevented for all subjects.

The payment system using the TOE as its component must be able to verify

a) the authenticity of the memory cards used for payments and
b) the correct debiting of an amount from the memory cards used for payments.

The verification is performed by the TOE, utilizing the Eurochip algorithm in itself. In order to prevent an attacker to generate a counterfeiting device, which could be used for payments, the TOE

a) must not reveal the algorithm,
b) must not allow modification of the algorithm, and
c) must not allow using the algorithm or its output for other purposes than verification of authenticity.

The functions F1, F2 and F3 provide this security functionality. Therefore the functions F1, F2 and F3 are appropriate for the intended method of use.

3.4 Security mechanisms

The security mechanisms (SEMs) realizing the SEFs are:

M1 *A manipulation of run-time (variable)16-bit read/write addresses so that they can access the EEPROM area only and never the RAM or ROM areas* (for target hardware memory topology reasons an access to ROM and EEPROM requires a 16-bit address).
The TOE includes also fixed (hardcoded) machine instructions that contain 16-bit addresses to access parts of the ROM, but never the algorithm. Because of being hardcoded, they cannot be changed.
All variable addresses (stored in RAM) that can point to the RAM, ROM and EEPROM address spaces are used for reading and writing purpose only by specific routines. These routines compute regularly an address in no other than the EEPROM area, and then use the computed address by

a read or write action. The read and write actions for all EEPROM addresses are coded in the TOE. Any address which points irregularly to memory outside EEPROM is converted into an address inside the EEPROM area in order to avoid ROM access.

M2 *An internal comparison of computed result (computed by algorithm in TOE) and expected result (computed by the Eurochip card).* This is to ensure using the algorithm for verification purpose only.

M3 *Destruction of all temporary data produced by the algorithm.* Only a binary information is passed as return value. This is to prevent any other part of TOE or TOE-external to use the temporary results of the algorithm.

M3a *Destruction of all temporary data during start-up.* All possible information left in the RAM memory by the previous execution of TOE is cleared unconditionally in start-up. This is to prevent any other part of TOE or TOE-external to use the temporary results of the algorithm in case of reset of power-down occurring during execution of the algorithm, mechanisms or security measures immediately following the algorithm.

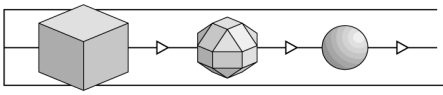
M4 C*heck that the return address path is correct before returning to calling routine.* Incorrect return address path causes a software reset of the TOE. This is to ensure using the algorithm successfully with continuing of the legal verification processing routine only.
The algorithm is entered legally from one SAPL routine, which performs verification of Eurochip card authentication, and the execution returns to that routine after execution of the algorithm.
This mechanism does not prevent calling or jumping to the algorithm from anywhere in TOE or TOE-externals, but it ensures that execution continues either from the only legal address or from TOE start-up address.

The SEFs and the SEMs are also supported by the following static security measures that are realized in software.

SM1 *The algorithm is realized as sequential code with no subroutines, and followed byM2, M3 and M4 sequentially.* This is to ensure that parts of the algorithm cannot be executed without executing the mechanisms M2, M3 and M4, and that all the mechanisms M2, M3 and M4 are executed after executing the algorithm or a part of it.

SM3 *The TOE (the ROM code) contains no byte patterns which could be used to read, test or modify the algorithm.* This is to prevent an executable code in RAM or EEPROM to read or modify the algorithm by means of calling or jumping to a random location in ROM code.

The mapping of



a) the mechanisms to functions and
b) the function-mechanism pairs to the security objectives and threats

is illustrated in the table 2 below. For example, T2 is countered by F2, using M1 and SM3, thus satisfying SO2.

	SO1 / T1	SO2 / T2	SO3 / T3
F1	M1, SM3		
F2		M1, SM3	
F3			M2, M3, M4, SM1

Table 2. Mapping of functions, mechanisms, additional security measures, security objectives and threats.

Readout of the algorithm is prevented by

a) M1, which prevents run-time addresses to be used for reading the algorithm,
b) SM3, which prevents any TOE-external or TOE-internal code to use parts of TOE for reading or comparing the memory area that contains the algorithm, and
c) supporting technical security measure, which prevents any code in RAM or EEPROM to read ROM area by the machine instructions.

Modification of the algorithm is prevented by

a) M1, which prevents run-time addresses to be used for writing the algorithm,
b) SM3, which prevents any TOE-external or TOE-internal code to use parts of TOE for writing the memory area that contains the algorithm,
c) supporting technical security measure, which prevents any code in RAM or EEPROM to write ROM area by the machine instructions, and
d) supporting technical security measure, which prevents modification of ROM memory.

Using the algorithm for other purposes than verification of authentication is prevented by

a) M2, which prevents using the algorithm or a part of it for any purpose other than verification of authentication result,
b) M3, which prevents any other part of TOE (or TOE-external) to get access to the temporary results of the algorithm,
c) M4, which prevents using the algorithm or a part of it without either continuing the intended and valid execution of TOE or restarting the entire TOE, and
d) SM1, which prevents using the algorithm or a part of it without usage of mechanisms M2, M3 and M4.

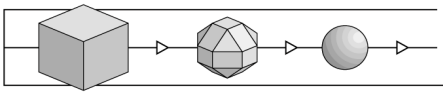
The security measure SM3 is not implemented directly in any component of the TOE, and it cannot be assured by hierarchical design methods. It is rather a verified characteristic of the complete TOE, influenced by the iterative implementation of the entire TOE. Therefore SM3 cannot be identified in the design documentation of the TOE, but it can be verified by inspecting the TOE.

3.5 Minimum strength of mechanisms

The *claimed* minimum strength of the TOE mechanisms M1 - M4 (see chapter 3.4) is medium.

3.6 Target evaluation level

The target evaluation level is E3 according to ITSEC.

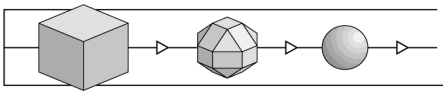


(This page is intentionally left blank.)

4 Remarks and Recommendations concerning the Certified Object

22 The statements given in chapter 2 are to be considered as the outcome of the evaluation.

23 The Certification Body has no further information or recommendations for the user.



(This page is intentionally left blank.)

5 Security Criteria Background

24 This chapter gives a survey on the criteria used in the evaluation and its different metrics.

5.1 Fundamentals

25 In the view of ITSEC security is given if there is sufficient assurance that a product or system meets its security objectives.

26 The security objectives for a product or system are a combination of requirements for

- confidentiality
- availability
- integrity

of certain data objects. The security objectives are defined by a vendor or developer for his product and by the user for his (installed) system.

27 The defined security objectives are exposed to *threats*, i.e. loss of confidentiality, loss of availability and loss of integrity of the considered data objects.

28 These threats become real, when subjects read, deny access to or modify data without authorisation.

29 Security (enforcing) functions provided by the considered product or system are intended to counter these threats.

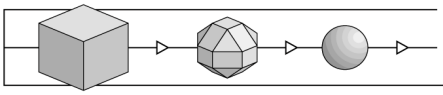
30 There are two basic questions:

- Do the security functions operate correctly?
- Are they effective?

Thus, an adequate assurance that the security objectives are met can be achieved if correctness and effectiveness have been evaluated.

5.2 Assurance level

31 An evaluation can only be performed with limited resources, especially limited time. Thus, the depth of an evaluation is always limited. On the other hand, it is not reasonable to perform an evaluation with extremely high resources when there is only need for low level security - and vice versa.



32 Therefore, it is reasonable to define a metric of assurance levels based on depth of the evaluation and resources needed. In ITSEC six assurance levels are given for the evaluation of correctness and effectiveness. E1 is the lowest, E6 the highest level.

33 Thus, the trustworthiness of a product or system can be “measured” by such assurance levels.

34 The following excerpt from the ITSEC shows which aspects are covered during the evaluation process and which depth of analysis corresponds to the assurance levels.

35 The enumeration contains certain requirements as to correctness and gives a first idea of the depth of the corresponding evaluation (“TOE” is the product or system under evaluation):

E1 “At this level there shall be a security target and an informal description of the architectural design of the TOE. Functional testing shall indicate that the TOE satisfies its security target.”

E2 “In addition to the requirements for level E1, there shall be an informal description of the detailed design. Evidence of functional testing shall be evaluated. There shall be a configuration control system and an approved distribution procedure.”

E3 “In addition to the requirements for level E2, the source code and / or hardware drawings corresponding to the security mechanisms shall be evaluated. Evidence of testing of those mechanisms shall be evaluated.”

E4 “In addition to the requirements for level E3, there shall be an underlying formal model of security policy supporting the security target. The security enforcing functions, the architectural design and the detailed design shall be specified in a semiformal style.”

E5 “In addition to the requirements for level E4, there shall be a close correspondence between the detailed design and the source code and / or hardware drawings.”

E6 “In addition to the requirements for level E5, the security enforcing functions and the architectural design shall be specified in a formal style, consistent with the specified underlying formal model of security policy.”

36 Effectiveness aspects have to be evaluated according to the following requirements identical for each level E1 to E6 :

“Assessment of effectiveness involves consideration of the following aspects of the TOE:

a) the suitability of the TOE's security enforcing functions to counter the threats to the security of the TOE identified in the security target;

b) the ability of the TOE's security enforcing functions and mechanisms to bind together in a way that is mutually supportive and provides an integrated and effective whole;

c) the ability of the TOE's security mechanisms to withstand direct attack;

d) whether known security vulnerabilities in the *construction* of the TOE could in practice compromise the security of the TOE;

e) that the TOE cannot be configured or used in a manner which is insecure but which an administrator or end-user of the TOE would reasonably believe to be secure;

f) whether known security vulnerabilities in the *operation* of the TOE could in practice compromise the security of the TOE."

5.3 Security Functions and Security Mechanisms

37 Typical examples for security functions are *Identification and Authentication* (of subjects), *Access Control*, *Accounting* and *Auditing*, *(Secure) Data Exchange*. Such security functions can be implemented in IT products and systems.

38 Functionality classes are formed by grouping a reasonable set of security functions.

Example: The functionality class F-C2 covers the generic headings *Identification and Authentication*, *Access Control*, *Accounting* and *Auditing*, and *Object Reuse*. This class is typical for many commercial operating systems.

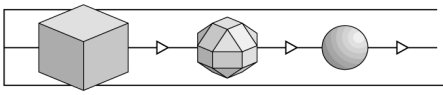
39 For every security function there are many ways of implementation:

Example: The function *Identification and Authentication* can be realised by a password procedure, usage of chipcards with a challenge response scheme or by biometrical algorithms.

40 The different implementations are called *(security) mechanisms* of the security function *Identification and Authentication*.
For other security functions the term mechanism is used similarly.

41 The rated ability of a security mechanism to counter potential direct attacks is called *strength* of (this) mechanism.

42 In ITSEM two types of mechanisms are considered: type B and type A.



Type B "A *type B mechanism* is a security mechanism which, if perfectly conceived and implemented, will have no weaknesses. A type B mechanism can be considered to be impregnable to direct attack regardless of the level of resources, expertise and opportunity deployed. A potential example of a type B mechanism would be access control based on access control lists: if perfectly conceived and implemented, this type B mechanism cannot be defeated by direct attack. However, these type B mechanisms can be defeated by indirect attacks which are the subject of other effectiveness analyses."

Considering direct attacks only, type B mechanisms cannot be defeated.

Type A "A *type A mechanism* is a security mechanism with a potential vulnerability in its algorithm, principles or properties, whereby the mechanism can be overcome by the use of sufficient resources, expertise and opportunity in the form of a direct attack. An example of a type A mechanism would be an authentication program using a password: if the password can be guessed by attempting all possible passwords in succession, the authentication mechanism is of type A. Type A mechanisms often involve the use of a "secret" such as a password or cryptographic key."

"All type A mechanisms ... have a strength, which corresponds to the level of resources, expertise and opportunity required to compromise security by directly attacking the mechanism."

43 How is the strength for type A mechanisms defined?

"All critical security mechanisms (i.e. those mechanisms whose failure would create a security weakness) are assessed for their ability to withstand direct attack. The minimum strength of each critical mechanism shall be rated either *basic*, *medium* or *high*."

basic "For the minimum strength of a critical mechanism to be rated *basic* it shall be evident that it provides protection against random accidental subversion, although it may be capable of being defeated by knowledgeable attackers."

medium "For the minimum strength of a critical mechanism to be rated *medium* it shall be evident that it provides protection against attackers with limited opportunities or resources."

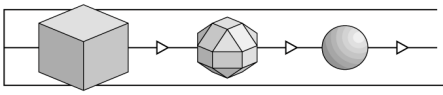
high "For the minimum strength of a critical mechanism to be rated *high* it shall be evident that it could only be defeated by attackers possessing a high level of expertise, opportunity and resources, successful attack being judged to be beyond normal practicability."

6 Annex

6.1 Glossary

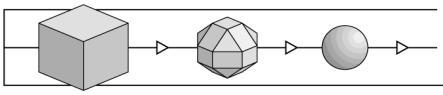
This glossary provides descriptions of the expressions used in this brochure, but does not guarantee their completeness or general validity. The term *security* here is always used in the context of information technology.

Accreditation	– A process to confirm that an evaluation facility complies with the requirements stipulated by the DIN EN 45001 standard. Accreditation is performed by an *accreditation body.* Accreditations from bodies represented in the German Accreditation Council (DAR) are generally recognised. – Result of an accreditation procedure.
Availability	Classical security objective: Data should always be available to authorised persons, i.e. this data should not be made inaccessible by unauthorised persons and should not be rendered unavailable due to technical defects.
Certificate	Summary representation of a certification result, issued by the certification body.
Certification	Independent confirmation of the correctness of an evaluation. This term is also used to describe the overall process consisting of evaluation, monitoring and subsequent issue of certificates and certification reports.
Certification body	An organisation which performs certifications (s. also „Trust centre“ for a second meaning).
Certification ID	Code designating a certification process.
Certification report	Report on the object, procedures and results of certification; this report is issued by the certification body.
Certification scheme	A summary of all principles, regulations and procedures applied at a certification body.
Certifier	Employee at a certification body authorised to carry out certification and to monitor evaluations.
Common Criteria	Security criteria derived from the US Orange Book / Federal Criteria, European ITSEC and Canadian CTCPEC, and intended to form an internationally accepted security standard.
Confidentiality	Classical security objective: Data should only be accessible to authorised persons.



Confirmation Body	Body that issues security confirmations in accordance with SiG and SigV for technical components (suitability) and trust centres (implementation of security concepts)
debisZERT	Name of the debis IT Security Services Certification Scheme.
Digital Signature Ordinance - SigV	Official regulations concerning the implementation of the German Signature Law.
EN 45000	A series of European standards applicable, in particular, to evaluation facilities and certification bodies.
Evaluation	Assessment of a product, system or service against defined security criteria and security standards.
Evaluation facility	The organisational unit which performs evaluations.
Evaluation level	Refer to „Security level“.
Evaluation report	Individual evaluation report or evaluation technical report.
Evaluation technical report	Final report written by an evaluation facility on the procedure and results of an evaluation (abbreviated as „ETR“ in the ITSEC context).
Evaluator	Person in charge of an evaluation at an evaluation facility.
Individual evaluation report	Report written by an evaluation facility on individual evaluation aspects as part of an evaluation.
Initial certification	Initial certification of a product, system or service.
Integrity	Classical security objective: Only authorised persons should be capable of modifying data.
IT component	A discrete part of an IT product or IT system.
IT product	Software and / or hardware which can be procured from a supplier (manufacturer, distributor).
IT service	A service related to the support of IT products and IT systems.
IT system	– A inherently functional combination of IT products. – (ITSEC:) A real installation of IT products with a known operational environment.
ITSEC	Information Technology Security Evaluation Criteria: European de facto standard for the evaluation of IT products and IT systems.
ITSEM	Information Technology Security Evaluation Manual. This manual on ITSEC applies in particular to evaluation processes.

Licence (personal)	Confirmation of a personal qualification (in the context of debisZERT here).
Licensing	Evaluation of organisation and qualification of an evaluation facility with respect to an intended licence agreement.
Licence agreement	An agreement between an evaluation facility and a certification body specifying procedures and responsibilities for evaluation and certification.
Manufacturers' laboratory	An organisational unit belonging to the manufacturer of a product /system or the supplier of a service, charged with performing evaluation of that product, system or service.
Milestone plan	A project schedule for the implementation of evaluation and certification processes.
Monitoring	Procedure implemented by the certification body in order to check whether an evaluation is performed correctly (compliance with criteria, use of standard processes and appraisal techniques etc.).
Pre-certification	Confirmation of the results of a preliminary investigation of a product-specific or process-specific security standard or a security-related tool (with a view to later certification).
Problem report	Report sent by an evaluation facility to the certification body and concerning special problems during evaluation.
Process ID	ID designating a certification or confirmation process within debisZERT.
Re-certification	Renewed certification of a new version following modification of a previously certified object; re-certification might also be required after a change of tools, production / delivery processes and security criteria.
Recognition (agreement)	Declaration and confirmation (of the equivalence of certificates and licences).
Regulation Authority (for Telecommunications and Post)	The authority responsible in accordance with §66 of the German Telecommunications Law (TKG).
Right of disposal	In this case: Authorisation to allow all inspections of a product, system or service as part of evaluation and certification.
Security certificate	Refer to „Certificate".
Security confirmation	In debisZERT: A legally binding confirmation of security features extending beyond the scope of a certificate.



Security criteria	Normative document that may contain technical requirements for products, systems and services, but at least describes the evaluation of such requirements.
Security function	Functions of an IT product or IT system for counter-acting particular threats.
Security level	Many criteria sets (e.g. ITSEC, CC) define a metric to indicate various levels of security relating to different requirements for the object to be certified and the degree of detail needed during evaluation.
Security specification	Security-related functional requirements for products, systems and services.
Security standards	A joint expression encompassing security criteria and security specifications.
Service type	Particular type of service (DLB) offered by debisZERT.
Signature Law - SigG	§3 of legislation on Information and Communications Services Act (IuKDG).
Sponsor	A natural or legal person who (in this case) issues an order for certification or evaluation, and who must possess a sufficient right of disposal for the object requiring certification.
System accreditation	Procedure of accepting an IT system or IT service for usage (considered here from the perspective of adequate security) in a specific environment and / or application.
Trust centre	A centre which confirms the relationship between signature keys and persons by means of electronic certificates - such a centre is termed „certification body“ in the German Signature Law.
ZKA criteria	Security criteria used by the central credit committee (ZKA) in Germany

6.2 References

/ A00 / Lizenzierungsschema (Licensing Scheme), debisZERT, Version 1.0, 7.8.98

/ ALG / Anhang zu „Bekanntmachung zur digitalen Signatur nach Signaturgesetz und Signaturverordnung vom 09.02.98 im Bundesanzeiger Nr. 31 v. 14.02.98“, http: / / www.RegTp.de / Fachinfo / Digitale%20Signatur / start.htm

[Annex to „Official Announcement concerning the Digital Signature according to the German Signature Law and Signature Ordinance by February 9, 1998 published in Bundesanzeiger No. 31, February 14, 1998“]

/BSIG/ Gesetz über die Errichtung des Bundesamtes für Sicherheit in der Informationstechnik (BSI-Errichtungsgesetz - BSIG), BGBl. I. vom 17. Dezember 1990, Seite 2834 ff.

[Law on the Establishment of the German Information Security Agency, BGBl. I. from 17th December 1990, Page 2834]

/CC/ Common Criteria for Information Technology Security Evaluation, CCIB-98-026, CCIB-98-027, CCIB-98-027A, CCIB-98-028, Version 2.0, Mai 1998

/EBA/ Kriterien für die sicherheitstechnische Bewertung und Konstruktion von CIR-Netzkomponenten, Eisenbahn-Bundesamt, Version 1.0 vom 8.2.94

[Criteria for Security-Related Evaluation and Construction of CIR Network Components, Federal Railway Office, version 1.0 from 8.2.94]

/ITSEC/ Information Technology Security Evaluation Criteria (ITSEC), Version 1.2 (1991), ISBN 92-826-3004-8

/ITSEM/ Information Technology Security Evaluation Manual (ITSEM), Version 1.0 (1993), ISBN 92-826-7087-2

/IuKDG/ Gesetz zur Regelung der Rahmenbedingungen für Informations- and Kommunikationsdienste (Informations- and Kommunikationsdienste-Gesetz - IuKDG), BGBl. I. vom 28. Juli 1997, Seite 1872 ff.

[Information and Communication Services Act, BGBl. I. from 28th July 1997, Page 1872]

/JIL/ Joint Interpretation Library, Version 1.04, December 97

/Mkat12/ Maßnahmenkatalog nach §12 Abs. 2, Regulierungsbehörde für Telekommunikation und Post,
http://www.RegTp.de/Fachinfo/DigitalSign/start.htm

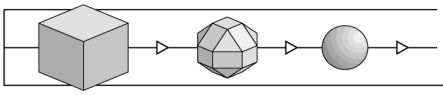
[Catalogue of Security Measures in accordance with §12 Abs. 2, Regulation Authority for Telecommunications and Post]

/Mkat16/ Maßnahmenkatalog nach §16 Abs. 6, Regulierungsbehörde für Telekommunikation und Post,
http://www.RegTp.de/Fachinfo/DigitalSign/start.htm

[Catalogue of Security Measures in accordance with §16 Abs. 6, Regulation Authority for Telecommunications and Post]

/SigG/ Article 3 of /IuKDG/

/SIGV/ Verordnung zur digitalen Signatur (Signaturverordnung - SigV), BGBl. I. vom 27.10.1997, Seite 2498 ff.



[Digital Signature Ordinance, BGBl. I. from 27th October 1997, Page 2498 ff.]

/TKG/ Telekommunikationsgesetz (TKG), BGBl. I. vom 25.7.1996, Seite 1120

[Telecommunications Act, BGBl. I. from 25.7.1996, Page 1120]

/V01/ Certificates in accordance with ITSEC/CC, Service type 1, Version 1.3E, September 17, 1998

/V02/ Confirmations for IT Products in accordance with the German Signature Law, Service type 2, Version 1.3E, September 10,1998

/V04/ Certificates recognised by the BSI, Service type 4, debisZERT, Version 1.3E, 5.8.98

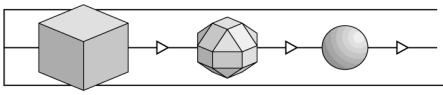
/Z01/ Certification Scheme, debis IT Security Services, Version 1.3E, 5.8.98

/Z02/ Certified IT Products, Systems and Services, debisZERT, Release 2, October 1998]

6.3 Abbreviations

AA	Work instructions
AIS	Request for an interpretation of security criteria
BSI	Bundesamt für Sicherheit in der Informationstechnik (German Information Security Agency)
BSIG	Act on the Establishment of the BSI
CC	Common Criteria for Information Technology Security Evaluation
CTCPEC	Canadian Trusted Computer Products Evaluation Criteria
DAR	Deutscher Akkreditierungsrat (the German Accreditation Council)
DBAG	Deutsche Bahn AG (the Federal German Railways Inc.)
debisZERT	The debis IT Security Services Certification Scheme
DEKITZ	Deutsche Akkreditierungsstelle für Informations- und Kommunikationstechnik (the German Accreditation Body for Information and Telecommunication Technology)
DLB	Service type
EBA	Eisenbahn-Bundesamt (the Federal German Railway Office)

ETR	Evaluation Technical Report
IT	Information technology
ITSEC	IT Security Evaluation Criteria
ITSEM	IT Security Evaluation Manual
IuKDG	German Information and Communication Services Act
LG	Management Board
SigG	German Digital Signature Act
SigV	German Signature Ordinance
TKG	German Telecommunications Act
TOE	Target of Evaluation
ZKA	Zentraler Kreditausschuß (German Central Credit Committee)
ZL	Head of the Certification Body
ZZ	Person in charge of a certification procedure (responsible certifier)



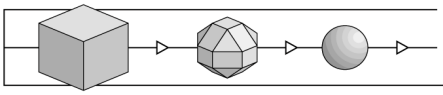
(This page is intentionally left blank.)

7 Re-Certification

44 When a certified object has been modified, a re-certification can be performed in accordance with the rules of debisZERT. The annexes to this chapter 7 (ordered by date of issuance) describe the type of modification, the new product version and the certification status.

45 If current findings in the field of IT security affect the security of a certified object, a technical annex to this certification report can be issued.

46 Re-certification and new technical annexes will be announced in the brochure /Z02/, also published on WWW.



End of initial version of the certification report.