CERTIFICATION REPORT

Certification file:	**TUVIT-DSZ-CC-9212**
Product / system:	signature creation device DataSIGN(tm) Security Token, Version 1.0
Product manufacturer:	First Data Corporation 10825 Farnam Drive, C-18 Omaha, NE 68154-3277, USA
Customer:	see above
Evaluation facility:	TÜViT, evaluation body for IT security
Evaluation report:	*Version 3 as of 2005-11-17* Document-number: *20568616_TÜV_023.3* Authors: Marcus Krechel, Dr. Wolfgang Hampe-Neteler
Result:	EAL4
Evaluation stipulations:	one (see chapter 10)
Certifier:	Dr. Christoph Sutter
Certification stipulations:	one (see chapter 11)

Essen, 2005-12-06

Dr. Ernst-Hermann Gruschwitz Dr. Christoph Sutter

Contents

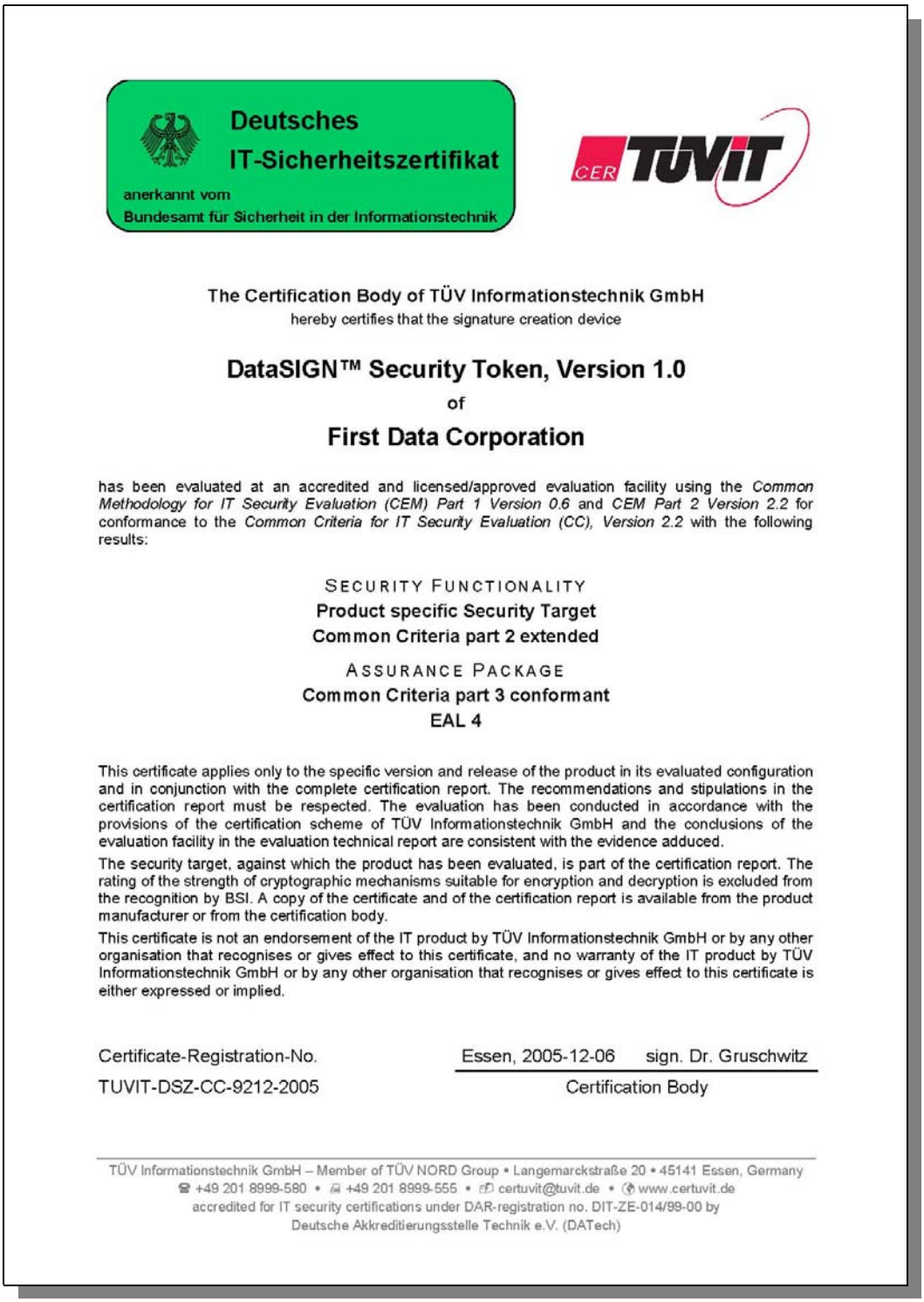
Part A

Certificate and Background of the Certification

Part A presents a copy of the issued certificate and summarizes

- information about the certification body,
- the certification procedure, and
- the performance of evaluation and certification.

1 The Certificate



Deutsches
IT-Sicherheitszertifikat
anerkannt vom
Bundesamt für Sicherheit in der Informationstechnik

CER TÜViT

The Certification Body of TÜV Informationstechnik GmbH
hereby certifies that the signature creation device

DataSIGN™ Security Token, Version 1.0
of
First Data Corporation

has been evaluated at an accredited and licensed/approved evaluation facility using the *Common Methodology for IT Security Evaluation (CEM) Part 1 Version 0.6* and *CEM Part 2 Version 2.2* for conformance to the *Common Criteria for IT Security Evaluation (CC), Version 2.2* with the following results:

SECURITY FUNCTIONALITY
Product specific Security Target
Common Criteria part 2 extended

ASSURANCE PACKAGE
Common Criteria part 3 conformant
EAL 4

This certificate applies only to the specific version and release of the product in its evaluated configuration and in conjunction with the complete certification report. The recommendations and stipulations in the certification report must be respected. The evaluation has been conducted in accordance with the provisions of the certification scheme of TÜV Informationstechnik GmbH and the conclusions of the evaluation facility in the evaluation technical report are consistent with the evidence adduced.

The security target, against which the product has been evaluated, is part of the certification report. The rating of the strength of cryptographic mechanisms suitable for encryption and decryption is excluded from the recognition by BSI. A copy of the certificate and of the certification report is available from the product manufacturer or from the certification body.

This certificate is not an endorsement of the IT product by TÜV Informationstechnik GmbH or by any other organisation that recognises or gives effect to this certificate, and no warranty of the IT product by TÜV Informationstechnik GmbH or by any other organisation that recognises or gives effect to this certificate is either expressed or implied.

Certificate-Registration-No.
TUVIT-DSZ-CC-9212-2005

Essen, 2005-12-06 sign. Dr. Gruschwitz
Certification Body

TÜV Informationstechnik GmbH – Member of TÜV NORD Group • Langemarckstraße 20 • 45141 Essen, Germany
☎ +49 201 8999-580 • 🖷 +49 201 8999-555 • certuvit@tuvit.de • www.certuvit.de
accredited for IT security certifications under DAR-registration no. DIT-ZE-014/99-00 by
Deutsche Akkreditierungsstelle Technik e.V. (DATech)

2 Certification Body – CERTÜVIT

CERTÜViT, the Certification Body of *TÜV Informationstechnik GmbH*[1] – Member of TÜV NORD Group – was established in 1998 and offers a variety of services in the context of security evaluation and validation.

CERTÜViT was accredited in September 1999 for certification of IT security products according to ITSEC and Common Criteria by *Deutsche Akkreditierungsstelle für Informations- und Telekommunikationstechnik (Dekitz)* now *Deutsche Akkreditierungsstelle Technik e.V. (DATech), Frankfurt/Main* under DAR-registration no. DAT-ZE-014/99-01 and performs its projects under a quality management system certified against ISO 9001 by *Germanischer Lloyd, Hamburg*.

CERTÜViT is accredited by *Bundesamt für Sicherheit in der Informationstechnik*[2] to issue the "German IT Security Certificate" which is recognised by BSI as equivalent to the "German IT Security Certificate" of BSI.

3 Specifications of the Certification Procedure

The certification body conducts the certification procedure according to the criteria laid down in the following:

- DIN EN 45011
- TÜViT Certification Scheme
- TÜViT Certification Conditions
- Regulations on the "German IT Security Certificate" issued by the BSI and accepted in the contract of BSI and TÜViT as of November 20, 2002.
- Common Criteria for Information Technology Security Evaluation (CC) part 1-3, version 2.2, January 2004.
- Common Methodology for Information Technology Security Evaluation (CEM) part 1, version 0.6, January 1997.
- Common Methodology for Information Technology Security Evaluation (CEM) part 2, version 2.2, January 2004.
- Application Notes and Interpretations of the Scheme (AIS), published by BSI.

[1] in the following termed shortly TÜViT

[2] in the following termed shortly BSI

4 Recognition Agreements

In order to avoid multiple certification of the same product by different certification bodies a mutual recognition of IT security certificates – as far as such certificates are based on ITSEC or CC – under certain conditions was agreed. CERTÜViT certificates are German IT Security Certificates recognized by BSI – the national German certification body in international agreements – to be equivalent to its own certificates but they are not part of these international agreements.

4.1 CC - Certificates

An arrangement (Common Criteria Arrangement) on the mutual recognition of certificates based on the CC evaluation assurance levels up to and including EAL4 was signed between the national participants of Australia and New Zealand, Austria, Canada, Czech Republic, Finland, France, Germany, Greece, Hungary, India, Israel, Italy, Japan, The Netherlands, Norway, Republic of Singapore, Spain, Sweden, Turkey, United Kingdom, and the United States.

4.2 ITSEC/CC - Certificates

The SOGIS-Agreement on the mutual recognition of certificates based on ITSEC was signed by the national bodies of Finland, France, Germany, Greece, Italy, The Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and the United Kingdom. The arrangement on the mutual recognition of IT security certificates based on the CC was extended by these participants up to and including the evaluation assurance level EAL7.

5 Performance of Evaluation and Certification

The certification body monitors each individual evaluation to ensure uniform procedures, interpretations of the criteria, and ratings. The signature creation device *DataSIGN(tm) Security Token, Version 1.0* has undergone the certification procedure at TÜViT certification body. It was an initial certification.

The evaluation of the signature creation device *DataSIGN(tm) Security Token, Version 1.0* was conducted by the evaluation body for IT-security of TÜViT and concluded on November 17, 2005. The TÜViT evaluation facility is recognised by BSI.

The sponsor as well as the developer is First Data Corporation. Distributor of the product is First Data Corporation.

The certification was concluded with

- the comparability check and
- the preparation of this certification report.

This work was concluded on December 6, 2005. The confirmation of the evaluation assurance level (EAL) only applies on the condition that

- all stipulations regarding generation, configuration and operation, as given in part B of this report, are observed,
- the product is operated – where indicated – in the environment described.

This certification report applies only to the version of the product indicated here. The validity of the certificate can be extended to cover new versions and releases of the product, provided the applicant applies for re-certification of the modified product, in accordance with the procedural requirements, and provided the evaluation does not reveal any security deficiencies.

With regard to the meaning of the evaluation assurance levels (EAL) and the strength of function (SoF), please refer to part C of this report.

6 Publication

The following Certification Results consist of pages B-1 to B-20. The product DataSIGN(tm) Security Token, Version 1.0 will be included in the BSI list of certified products which is published at regular intervals (e. g. in the Internet at http://www.bsi.bund.de) and the TÜViT certification lists (http://www.certuvit.de).

Further copies of this certification report may be ordered from the sponsor of the product. The certification report may also be obtained in electronic form at the internet address of CERTÜViT as stated above.

Part B

Certification Result

The following results represent a summary of

- the security target of the sponsor for the target of evaluation,
- the relevant evaluation results from the evaluation facility, and
- complementary notes and stipulations of the certification body.

Contents of the Certification Result

1 Executive Summary

1.1 Target of Evaluation and Evaluation Background

The target of evaluation (TOE) is the signature creation device (smart card) ***DataSIGN(tm) Security Token Version 1.0***[3] and consists of two main components:

a.) Infineon smart card security controller SLE66C42P / m1495a14 and

b.) DataSIGN(tm) Security Token embedded software which combines a smart card operating system and all required software functionality.

The smart card IC, the Infineon SLE66C42P / m1495a14 was certified on May 31st, 2005 by TUVIT under certification ID: TUVIT-DSZ-CC-9243-2005 at the level EAL5 augmented by ALC_DVS.2, AVA_MSU.3, and AVA_VLA.4 [TUVIT9243]. The evaluation and certification results from the smart card IC certification have been considered in this certification.

The DataSIGN(tm) Security Token provides an electronic signature (SHA-1/ECDSA 163 bit) on any data presented to it. Before signing, the TOE automatically adds *PIN Status Data* to the data to be signed. The *PIN Status Data* can have the following values:

- 0x00: a valid PIN was presented just before the signature,
- 0x01: a valid PIN was presented one ore more signatures before,
- 0x02: no valid PIN was presented,
- 0x03: the last PIN verification failed,

and provides information for the recipient of the signed message about the fact, that there was a valid PIN presented just before or a valid PIN was presented some signatures before or no PIN was presented any time before or the last PIN verification before signature had failed.

Furthermore, the TOE contains the optional feature of adding *User Data* (like a bank account number) to the data to be signed before signature. *User Data* is stored on the TOE securely, i. e. either it can be changed only after successful verification of the PIN or it is configured to be unchangeable. Depending on the configuration, *User Data* is inserted into the data to be signed. *User Data* is stored as XML formatted Tags. If *User Data* is allowed to be added, only the appropriate data of the same XML objects will be replaced in the data to be signed.

[3] In the following shortly termed DataSIGN(tm) Security Token.

In conclusion, the TOE will provide a signature over the following set of data:

- data to be signed, provided from the user,
- PIN Status Data, and
- optionally, User Data stored in the TOE.

The TOE was evaluated against the claims of the Security Target[4] [ST] (attached in part D) by "*evaluation body of TÜV Informationstechnik GmbH*" (TÜViT). The evaluation was completed on November 17, 2005. TÜViT's evaluation body is recognised by BSI.

1.2 Assurance Package

The TOE security assurance requirements are based entirely on the assurance components and classes defined in Part 3 of the Common Criteria (see part C of this report or [CC] Part 3 for details). The TOE meets the assurance requirements of assurance level EAL 4 (Evaluation Assurance Level 4 – methodically designed, tested, and reviewed).

1.3 Strength of Functions

The TOE's strength of functions is rated "high" (SOF-high). The strength of functions rating does not include cryptographic algorithms for encryption and decryption. For more details see also chapter 9 of this report.

1.4 Functionality

Except the functional requirement FCS_RND.1 (Quality metric for random numbers), the TOE security functional requirements were taken from CC part 2 (i. e. the set is CC part 2 extended) [CC]. They can be categorized in the following five categories:

1. cryptographic support,
2. user data protection,
3. identification and authentication,
4. security management, and
5. protection of the TSF.

[4] hereinafter called ST

Chapter 9 lists the security functional requirements in more detail. They are met by five suitable TOE security functions (TSF):

TSF	Short Description
1. KeyGenerate	Provides public/private key generation for the elliptic curve algorithm ECDSA (curve K-163) with 163 bit. The key generation can only be performed once during power on self test (POST).
2. UserAuthentication	Provides user authentication using an 8 Byte PIN. If PIN change is possible according the configuration, the old 8 Byte PIN must be entered before PIN change is allowed. The PIN authentication functionality is blocked after a configurable number (1 to 255) of wrong PIN entries.
3. SignatureCreation	Provides a signature (SHA-1/ECDSA 163 bit) over supplied data, the PIN Status Data, and optionally the User Data.
4. DataAccess	Import/export of User Data, Public Key, and TOE Status is allowed depending upon the token state and the configuration settings. In Error state no access between subjects and objects is allowed.
5. AssetProtection	When the defined number of unsuccessful authentication attempts has been met or surpassed the public/private key pair and the PIN is deleted. On detection of a physical attack access is denied until the next power up.

Table 1: TOE Security Functions

A more detailed description of the TOE security functions can be found in chapter 6 of the public ST, which is attached as part D of this certification report.

1.5 Summary of Threats and Organisational Security Policies (OSPs)

The following assets, threats and attackers (User, Owner, Other attacker) are identified in section 3.3 of the public security target [ST]:

Asset	Threat	Description
Signature	T.SignatureAnalyse	The Other attacker could forge digital signatures using a crypto analytical and mathematical attack.
PIN Status Data	T.PINStatusChange	The User and Other attackers could manipulate the PIN status field using a direct hardware attack.

Asset	Threat	Description
Private Signature Key	T.KeyPrivReadOut	The Other attacker could read out the Private Key using a direct hardware attack.
	T.KeyPrivDPADFA	The Other attacker could retrieve the users Private Key by performing a DPA or DFA to the TOE.
	T.KeyPrivCalc	The Other attacker could recalculate the private key by using crypto analytical and mathematical means.
Public signature key	T.KeyPub	The Other attacker could alter the public key during export by accessing the data stream.
Configuration data	T.ConfDataChange	The User, Owner or the Other attacker could alter the configuration data in contradiction to the configuration of the TOE.
PIN and User Data	T.IDDataGuess	The User or the Other attacker could guess the PIN by trying one after the other.
	T.IDDataPINCalc	The Other attacker could recalculate the PIN (valid only in case the PIN is not user changeable and has been generated during production phase)
	T.IDDataAlterate	The User, the Owner or the Other attacker could alter user data by accessing the TOE in an unauthorized way.

Table 2: Threats

No organisational security policies are defined.

1.6 Special Configuration Requirements

The TOE is delivered in one fixed configuration and no further generation takes place after delivery to the customer.

1.7 Assumptions about the Operating Environment

The life-cycle of the TOE can be divided into two phases, whereby delivery is between phase 1 and 2:

1. TOE Development & Manufacturing Phase comprises
   - Firm- & Software Development
   - Chip Design (HW)
   - Chip Production & Configuration

2. TOE Usage Phase comprises
   - Packaging
   - Usage
   - End of Life

The TOE operating environment in the Usage Phase is highly variable. It is assumed that the TOE and the TOE communication data can not be accessed in an unauthorised way (assumption A.EnvUsage). All usage and environmental assumptions can be found in chapter 4 of this certification report.

1.8 Independence of the Certifier

Within the last two years, the certifier did not render any consulting- or other services for the company ordering the certification and there was no relationship between them that might have an influence on his assessment.

The certifier did not participate at any time in test procedures for the product, which forms the basis of the certification.

1.9 Disclaimers

The certification results only apply to the version of the product indicated in the certificate and on the condition that all the stipulations are kept with regard to generation, configuration and operation as detailed in this certification report. This certificate is not an endorsement of the IT product by TÜV Informationstechnik GmbH or any other organisation that recognises or gives effect to this certificate, and no warranty of the IT product by TÜV Informationstechnik GmbH or any other organisation that recognises or gives effect to this certificate, is either expressed or implied.

2 Identification of the TOE

The Target of Evaluation (TOE) is the signature creation device *DataSIGN(tm) Security Token, Version 1.0* which comprises the hardware of the security controller and the embedded software.

The TOE delivery comprised the TOE itself in form of a module and the guidance ([ADM], [USR]) in form of encrypted electronic files as indicated in the table below.

No	Type	Identifier	Form of delivery
1	HW FW SW	**DataSIGN(tm) Security Token, Version 1.0** containing: • Security Controller SLE66C42P / m1495a14 • RMS Library V0.8 • STS V0.8 • DataSIGN(tm) soft-/firmware V2.0.3.0	module
2	DOC	Administrator Guidance for DataSIGN(tm) Security Token V1.0, version 1.1, 2005-08-18	electronic file
3	DOC	User Guidance For DataSIGN(tm) Security Token V1.0, version 1.1, 2005-08-18	electronic file

Table 3: Deliverables of the TOE

The TOE (item 1) is securely delivered by specific haulage companies from Infineon to the customer, generally the Card Issuer. The TOE can be identified by using the “GET STATUS” APDU command as described in section 3.3.8 of [USR].

The guidance documentation [ADM] and [USR] is send encrypted from First Data Corporation to the customer: [ADM] to the configurator (Infineon) and [USR] to the Card Issuer.

3 Security Policy

The TOE provides the security function policy *Access Controls SFP* depending on the TOE states (Configurator, User, Owner, and Error) to protect the sensitive objects (private and public signature keys, User Data, PIN, Configuration Data, and the PIN Status Data) and operations (import, export, and signature generation) of the TOE. The TOE states exclude each other, i. e they do not exist at the same time. Error state is possible for the whole life cycle. Configurator state is only possible during production while User and Owner states are only possible after production.

The main security policy of the TOE consist of allowing everybody to generate signatures and every signature contains the *PIN Status Data* indicating if a successful PIN authentication was done beforehand. (see also section 1.1)

A more detailed description of the security policy can be found in sections 5.1.1 and 5.1.1.4 of the public ST, which is attached as part D of this certification report.

4 Assumptions and Clarification of Scope

4.1 Usage Assumptions

The following usage assumptions are defined in [ST] and must be regarded when using the TOE:

Assumption	Description
A.PINChange	If an institution does not want the end user to change the PIN, the PIN change option during configuration should be set to “change once”. The institution should generate the PIN properly regarding length and alpha numerical characters, set it in the token, and communicate it to the end user using secure transmission. Therefore only the end user and the institution are assumed to know the PIN.
A.PINConfidentiality	The PIN is kept confidential by the user. The user therefore uses the TOE in a secure usage environment which enables the user to keep the PIN confidential, furthermore the user is informed about how to handle the PIN in a secure way.

4.2 Environmental Assumptions

The following environmental assumptions are defined in [ST] and must be regarded when using the TOE:

Assumption	Description
A.Development&Production	A secure production and configuration environment is assumed and the configurator is assumed to be trustworthy.
A.EnvUsage	The usage environment is secure in a way that the TOE and the TOE communication data can not be accessed in an unauthorized way.

4.3 Clarification of Scope

The Target of Evaluation (TOE) is the signature creation device *DataSIGN(tm) Security Token, Version 1.0*. The TOE will provide an electronic signature on any data presented to it, even if the PIN was not presented beforehand. Only in case the *PIN Status Data*, that is included in the signature, has the value 0x00, a valid PIN was presented just before the signature. See section 1.1 for more information on the *PIN Status Data*.

5 Architectural Information

The TOE comprises two major components: the Infineon smart card security controller SLE66C42P / m1495a14) and the DataSIGN(tm) Security Token embedded software which combines a smart card operating system and all required software functionality. The embedded software can be divided further into three subsystems as indicated in the following table:

FW-Subsystems	Description
Main Control	it handles the overall control of the firmware; includes the startup sequence, POST, I/O control and polling for the APDU commands
Command Functions	processes APDU commands
Common Functions	performs Elliptic Curve Cryptography (ECC), and provides utilities for subsystem communications and accessing low-level hardware features.

Table 4: Firmware Subsystems

For architectural information about the Infineon smart card security controller, see the corresponding certification report [TUVIT9243].

6 Documentation

The following documentation is provided with the product by the developer to the consumer as indicated in chapter 2 above:

- Administrator Guidance for DataSIGN(tm) Security Token V1.0, version 1.1, 2005-08-18 [ADM] (for the configurator – Infineon)
- User Guidance For DataSIGN(tm) Security Token V1.0, version 1.1, 2005-08-18 [USR] (for the card issuer)

7 IT Product Testing

The developer tested the TOE with the overall objectives to verify that the TOE satisfies all requirements specified in Functional Specification (FSP) and that it is a correct and complete implementation of the High Level Design (HLD) description.

The developers testing effort can be summarised in the following four aspects [ETR]:

TOE test configuration:

- The tests are performed with the chip DataSIGN(tm) Security Token, Version 1.0 and additionally in a special test configuration.

Testing approach:

- The developer's tests were conducted with the goal to confirm that the TOE meets the security functional requirements. Thereby, the developer's strategy was to test the TOE against the specification and all external interfaces (APDU commands) of security enforcing functions detailed in the functional specification.
- The tests cover the security functions defined in the FSP and ST and correspond to the HLD subsystems (including subsystem interfaces) and the FSP TSFI.

Amount of developer testing performed:

- Besides conformance testing, all possible APDU commands for the default configuration were tested.
- During acceptance testing, all 128 possible configurations (bit 1-7 of first byte of configuration data) were used for the test scenarios.
- With regard to the depth of testing, the tests ensure that the TSF have the effect as specified in the high-level design HLD.

Testing result:

- Overall the TSF have been tested systematically against the Functional Specification and the High Level Design.
- The developer tests demonstrate that the security functions perform as specified.
- All test results are positive and none is failed.

Tests of the evaluation body:

The independent testing of the evaluation body was performed in the developer’s testing environment. The same platforms and tools as for the developer tests were used.

The evaluator's objective regarding this aspect was to test the functionality of the TOE as described in the Functional Specification and the High Level Design, and to verify the developer's test results by conducting all of the developer's tests and additionally add independent tests. The tests include all security functions.

The results of the specified and conducted independent evaluator tests confirm the TOE functionality as described in the functional specification and the high level design. The TOE security functions were found to behave as specified.

The results of the developer tests, which have been repeated by the evaluator, matched the results of the developer.

The penetration testing according to AVA_VLA.2 was performed in the developer's testing environment considering all vulnerabilities found by the developer and evaluator. The same platforms and tools as for the developer tests were used.

The TOE is resistant against all attacks based on the level of a low attack potential.

The penetration testing conducted confirms that all vulnerabilities were considered and that the vulnerabilities identified are non-exploitable in the intended operational environment of the TOE, if taking into consideration all the measures the user is informed about.

8 Evaluated Configuration

The TOE *DataSIGN(tm) Security Token, Version 1.0* is delivered in one fixed configuration and no further generation takes place. Therefore, the evaluated configuration is identical to the TOE, which can be identified as described in chapter 2 of this certification report.

9 Results of the Evaluation

The Evaluation Technical Report [ETR] was provided by TÜViT's evaluation body according to the requirements of the Scheme, the Common Criteria [CC], the Methodology [CEM] and the Application Notes and Interpretations of the Scheme [AIS]. Especially, the following Application Notes and Interpretations of the Scheme were used in the present certification:

- [AIS 25], [AIS 26], and [AIS 37] for smart card IC specific methodology and
- [AIS 20] for the assessment of the random number generator.

The verdicts for the CC, part 3 assurance classes and components (according to EAL4 and the class ASE for the Security Target Evaluation) are summarised in the following table:

Assurance classes and components		Verdict
Security Target evaluation	**CC Class ASE**	PASS
TOE description	ASE_DES.1	PASS
Security environment	ASE_ENV.1	PASS
ST introduction	ASE_INT.1	PASS
Security objectives	ASE_OBJ.1	PASS
PP claims	ASE_PPC.1	PASS
IT security requirements	ASE_REQ.1	PASS
Explicitly stated IT security requirements	ASE_SRE.1	PASS
TOE summary specification	ASE_TSS.1	PASS
Configuration Management	**CC Class ACM**	PASS
Partial CM automation	ACM_AUT.1	PASS
Generation support and acceptance procedures	ACM_CAP.4	PASS
Problem tracking CM coverage	ACM_SCP.2	PASS
Delivery and operation	**CC Class ADO**	PASS
Detection of modification	ADO_DEL.2	PASS
Installation, generation, and start-up procedures	ADO_IGS.1	PASS
Development	**CC Class ADV**	PASS
Fully defined external interfaces	ADV_FSP.2	PASS
Security enforcing high-level design	ADV_HLD.2	PASS
Subset of the implementation of the TSF	ADV_IMP.1	PASS
Descriptive low-level design	ADV_LLD.1	PASS
Informal correspondence demonstration	ADV_RCR.1	PASS
Informal TOE security policy model	ADV_SPM.1	PASS
Guidance documents	**CC Class AGD**	PASS
Administrator guidance	AGD_ADM.1	PASS
User guidance	AGD_USR.1	PASS
Life cycle support	**CC Class ALC**	PASS
Identification of security measures	ALC_DVS.1	PASS
Developer defined life-cycle model	ALC_LCD.1	PASS
Well-defined development tools	ATE_TAT.1	PASS
Tests	**CC Class ATE**	PASS
Analysis of coverage	ATE_COV.2	PASS
Testing: high-level design	ATE_DPT.1	PASS
Functional testing	ATE_FUN.1	PASS
Independent testing – sample	ATE_IND.2	PASS
Vulnerability assessment	**CC Class AVA**	PASS
Validation of analysis	AVA_MSU.2	PASS
Strength of TOE security function evaluation	AVA_SOF.1	PASS
Independent vulnerability analysis	AVA_VLA.2	PASS

All assurance components were taken from [CC] part 3 and assessed with the verdict PASS. This includes that all evaluator action elements being part of the assurance components are also assessed with PASS. Therefore, the TOE as defined in the security target is considered to be Part 3 conformant.

Section 5.1 of the public ST, which is attached as part D of this certification report, lists the following TOE security functional requirements.

ID	**Class**/Component
FCS	**Cryptographic support**
FCS_CKM.1	Cryptographic key generation
FCS_CKM.4	Cryptographic key destruction
FCS_COP.1	Cryptographic operation
FCS_RND.1	*Quality metric for random numbers*
FDP	**User data protection**
FDP_ACC.2	Complete access control
FDP_ACF.1	Security attribute based access control
FDP_ETC.1	Export of user data without security attributes
FDP_ITC.1	Import of user data without security attributes
FIA	**Identification and authentication**
FIA_AFL.1	Authentication failure handling
FIA_UAU.1	Timing of authentication
FIA_UAU.7	Protected authentication feedback
FIA_UID.1	Timing of identification
FMT	**Security management**
FMT_MSA.1	Management of security attributes
FMT_MSA.2	Secure security attributes
FMT_MSA.3	Static attribute initialisation
FMT_SMF.1	Specification of management functions
FMT_SMR.1	Security roles
FPT	**Protection of the TSF**
FPT_AMT.1	Abstract machine testing
FPT_FLS.1	Failure with preservation of secure state
FPT_TST.1	TSF testing

Apart from *FCS_RND.1* (marked in italics in the table above) the security functional requirements were taken from [CC] part 2, i. e. the TOE is [CC] part 2 extended.

The evaluation performed in accordance to EAL4 has shown that the TOE security functional requirements are correctly realised by the TOE security functions. Thus, in realising these functional requirements, it is assured that the TOE will meet the security objectives claimed in the ST.

TSF1 (KeyGenerate), TSF2 (UserAuthentication), and TSF3 (SignatureCreation) fulfil the SOF-rating high (SOF-high). The strength of function rating for the RNG was performed according to class K3 and SOF-high of [AIS 20]. The strength of functions rating does not include cryptographic algorithms for encryption and decryption, like ECDSA in TSF3 (SignatureCreation).

The sponsor must advise the certification authority about any modification of the TOE or its guidance documentation. The certification authority will then check whether the certification results are still valid and, if necessary, initiate all further steps concerning a re-evaluation. The results of the evaluation are only applicable to the product "*DataSIGN(tm) Security Token, Version 1.0*". The validity can be extended to new versions and releases of the product or to chips from other production and manufacturing sites, provided the sponsor applies for re-certification of the modified product, in accordance with the procedural requirements, and the evaluation does not reveal any security deficiencies.

10 Evaluation Stipulations, Comments, and Recommendations

The Evaluation Technical Report [ETR] contains the following stipulation:

> The following requirement has to be considered by the developer: During the configuration of the TOE the firmware configuration data bit 5 (Random Number Generator) must be set to FIPS 186 based RNG (value 1).

The Evaluation Technical Report [ETR] contains the following comments and recommendations for the administrator and user:

> The configuration of the TOE is done by the administrator (configurator) by calling the SET CONFIGURATION command. Thereby, the firmware configuration data bit 5 (Random Number Generator) must be set to 1 (for FIPS 186 based) - only software based RNG is allowed for the TOE.
>
> There are no other recommendations & hints necessary for the user, except those provided by the guidance documents [ADM] and [USR].

11 Certification Stipulations and Notes

The stipulation and notes of the evaluation report (see chapter 10) are applicable.

12 Security Target

The public version of the security target [ST] for the signature creation device *DataSIGN(tm) Security Token, Version 1.0* is included in part D of this certification report.

13 Definitions

13.1 Acronyms

ADM	Administrator Guidance
CC	Common Criteria for Information Technology Security Evaluation (referenced to as [CC])
CEM	Common Methodology for Information Technology Security Evaluation (referenced to as [CEM])
CM	Configuration Management
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DOC	Documentation
DPA	Differential Power Analysis
EAL	Evaluation Assurance Level
EEPROM	Electrical Erasable and Programmable Read Only Memory
FSP	Functional Specification
HLD	High-level Design
HW	Hardware
IC	Integrated Circuit
IF	Interface
IGS	Installation, Generation and Start-up
OSP	Organisational Security Policy
PP	Protection Profile
RAM	Random Access Memory
RNG	Random Number Generator
ROM	Read Only Memory
SAR	Security Assurance Requirement
SF	Security Function
SFP	Security Function Policy
SFR	Security Functional Requirement
SIF	Sub-interface
SOF	Strength of Function
SPA	Simple Power Analysis
SS	Sub-system
ST	Security Target
SW	Software
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions

TSFI	TOE Security Function Interfaces
TSP	TOE Security Policy
USR	User Guidance
VLA	Vulnerability Analysis

13.2 Glossary

Augmentation – The addition of one or more assurance component(s) from Part 3 to an EAL or assurance package.

Extension – The addition to an ST or PP of functional requirements not contained in Part 2 and/or assurance requirements not contained in Part 3 of the CC.

Formal – Expressed in a restricted syntax language with defined semantics based on well-established mathematical concepts.

Informal – Expressed in natural language.

Object – An entity within the TSC that contains or receives information and upon which subjects perform operations.

Protection Profile – An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.

Security Function – A part or parts of the TOE that have to be relied upon for enforcing a closely related subset of the rules from the TSP.

Security Target – A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.

Semiformal – Expressed in a restricted syntax language with defined semantics.

Strength of Function – A qualification of a TOE security function expressing the minimum efforts assumed necessary to defeat its expected security behaviour by directly attacking its underlying security mechanisms.

SOF-basic – A level of the TOE strength of function where analysis shows that the function provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential.

SOF-medium – A level of the TOE strength of function where analysis shows that the function provides adequate protection against straightforward or intentional breach of TOE security by attackers possessing a moderate attack potential.

SOF-high – A level of the TOE strength of function where analysis shows that the function provides adequate protection against deliberately planned or organised breach of TOE security by attackers possessing a high attack potential.

Subject – An entity within the TSC that causes operations to be performed.

Target of Evaluation – An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.

TOE Security Functions – A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

TOE Security Policy – A set of rules that regulate how assets are managed, protected and distributed within a TOE.

TSF Scope of Control – The set of interactions that can occur with or within a TOE and are subject to the rules of the TSP.

14 Bibliography

[ADM] Administrator Guidance for DataSIGN(tm) Security Token V1.0, version 1.1, 2005-08-18

[AIS] Application Notes and Interpretations of the Scheme (AIS), published by BSI

[AIS 20] AIS 20, Version 1, as of 1999-12-02 including the document: “Functionality Classes and Evaluation Methodology for Deterministic Random Number Generators”, Version 2.0, 1999-12-02

[AIS 25] AIS 25, Version 2, as of 2002-07-29 including the CC supporting document: “The Application of CC to Integrated Circuits”, Version 1.2, 07’2002

[AIS 26] AIS 26, Version 2, as of 2002-08-06 including the CC supporting document: “Application of Attack Potential to Smartcards”, Version 1.1, 07’2002

[AIS 37] AIS 37, Version 1, as of 2002-07-29 including the CC supporting document: “Guidance for smartcard evaluation”, Version 1.1, 03’2002

[CC] Common Criteria for Information Technology Security Evaluation, Version 2.2, January 2004,
Part 1: Introduction and general model
Part 2: Security functional requirements
Part 3: Security assurance requirements

[CEM] Common Methodology for Information Technology Security Evaluation,
Part 1: Introduction and general model, version 0.6, revision 11.01.1997,
Part 2: Evaluation Methodology, Version 2.2, January 2004

[ETR] Evaluation Technical Report (ETR), TÜV Informationstechnik GmbH, version 3, 2005-11-17, document-number: 20568616_TÜV_023.3

[ST] DataSIGN(tm) Security Token, Version 1.0, EAL4 – SOF high, Security Target, Version 1.5, 2005-11-17

[TUVIT9243] Certification Report – TUVIT-DSZ-CC-9243-2005 for Infineon Smart Card IC (Security Controller) SLE66C82P / m1474a14, SLE66C42P / m1495a14, 2005-05-31

[USR] User Guidance For DataSIGN(tm) Security Token V1.0, version 1.1, 2005-08-18

Part C

Excerpts from the Criteria

The excerpts from the criteria are dealing with

- conformance results
- assurance categorization
- evaluation assurance levels
- strength of security function
- vulnerability analysis

CC Part 1:

Conformance results

„The conformance result indicates the source of the collection of requirements that is met by a TOE or PP that passes its evaluation. This conformance result is presented with respect to Part 2 (functional requirements), Part 3 (assurance requirements) and, if applicable, to a pre-defined set of requirements (e.g., EAL, Protection Profile).

The conformance result consists of one of the following:

> **Part 2 conformant** - A PP or TOE is Part 2 conformant if the functional requirements are based only upon functional components in Part 2.
>
> **Part 2 extended** - A PP or TOE is Part 2 extended if the functional requirements include functional components not in Part 2.

plus one of the following:

> **Part 3 conformant** - A PP or TOE is Part 3 conformant if the assurance requirements are based only upon assurance components in Part 3.
>
> **Part 3 extended** - A PP or TOE is Part 3 extended if the assurance requirements include assurance requirements not in Part 3.

Additionally, the conformance result may include a statement made with respect to sets of defined requirements, in which case it consists of one of the following:

> ***Package name* Conformant** - A PP or TOE is conformant to a pre-defined named functional and/or assurance package (e.g. EAL) if the requirements (functions or assurance) include all components in the packages listed as part of the conformance result.
>
> ***Package name* Augmented** - A PP or TOE is an augmentation of a predefined named functional and/or assurance package (e.g. EAL) if the requirements (functions or assurance) are a proper superset of all components in the packages listed as part of the conformance result.

Finally, the conformance result may also include a statement made with respect to Protection Profiles, in which case it includes the following:

> ***PP Conformant*** - A TOE meets specific PP(s), which are listed as part of the conformance result."

CC Part 3:

Assurance categorisation

The assurance classes, families, and the abbreviation for each family are shown in *Table 5*.

Assurance Class	Assurance Family	Abbreviated Name
Class ACM: Configuration management	CM automation	ACM_AUT
	CM capabilities	ACM_CAP
	CM scope	ACM_SCP
Class ADO: Delivery and operation	Delivery	ADO_DEL
	Installation, generation and start-up	ADO_IGS
Class ADV: Development	Functional specification	ADV_FSP
	High-level design	ADV_HLD
	Implementation representation	ADV_IMP
	TSF internals	ADV_INT
	Low-level design	ADV_LLD
	Representation correspondence	ADV_RCR
	Security policy modeling	ADV_SPM
Class AGD: Guidance documents	Administrator guidance	AGD_ADM
	User guidance	AGD_USR
Class ALC: Life cycle support	Development security	ALC_DVS
	Flaw remediation	ALC_FLR
	Life cycle definition	ALC_LCD
	Tools and techniques	ALC_TAT
Class ATE: Tests	Coverage	ATE_COV
	Depth	ATE_DPT
	Functional tests	ATE_FUN
	Independent testing	ATE_IND
Class AVA: Vulnerability assessment	Covert channel analysis	AVA_CCA
	Misuse	AVA_MSU
	Strength of TOE security functions	AVA_SOF
	Vulnerability analysis	AVA_VLA

Table 5: Assurance family breakdown and mapping

Evaluation assurance levels (chapter 6)

“The Evaluation Assurance Levels (EALs) provide an increasing scale that balances the level of assurance obtained with the cost and feasibility of acquiring that degree of assurance. The CC approach identifies the separate concepts of assurance in a TOE at the end of the evaluation, and of maintenance of that assurance during the operational use of the TOE.

It is important to note that not all families and components from Part 3 are included in the EALs. This is not to say that these do not provide meaningful and desirable assurances.

Instead, it is expected that these families and components will be considered for augmentation of an EAL in those PPs and STs for which they provide utility."

Evaluation assurance level (EAL) overview

„Table 6 represents a summary of the EALs. The columns represent a hierarchically ordered set of EALs, while the rows represent assurance families. Each number in the resulting matrix identifies a specific assurance component where applicable.

As outlined in the next section, seven hierarchically ordered evaluation assurance levels are defined in the CC for the rating of a TOE's assurance. They are hierarchically ordered inasmuch as each EAL represents more assurance than all lower EALs. The increase in assurance from EAL to EAL is accomplished by *substitution* of a hierarchically higher assurance component from the same assurance family (i. e. increasing rigour, scope, and/or depth) and from the *addition* of assurance components from other assurance families (i. e. adding new requirements).

These EALs consist of an appropriate combination of assurance components as described in chapter 2 of CC Part 3. More precisely, each EAL includes no more than one component of each assurance family and all assurance dependencies of every component are addressed.

While the EALs are defined in the CC, it is possible to represent other combinations of assurance. Specifically, the notion of "augmentation" allows the addition of assurance components (from assurance families not already included in the EAL) or the substitution of assurance components (with another hierarchically higher assurance component in the same assurance family) to an EAL. Of the assurance constructs defined in the CC, only EALs may be augmented. The notion of an "EAL minus a constituent assurance component" is not recognised by the CC as a valid claim. Augmentation carries with it the obligation on the part of the claimant to justify the utility and added value of the added assurance component to the EAL. An EAL may also be extended with explicitly stated assurance requirements."

Assurance Class	Assurance Family	Assurance Components by Evaluation Assurance Level						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Configuration Management	ACM_AUT				**1**	1	**2**	2
	ACM_CAP	**1**	**2**	**3**	**4**	4	**5**	5
	ACM_SCP			**1**	**2**	**3**	3	3
Delivery and Operation	ADO_DEL		**1**	1	**2**	2	2	**3**
	ADO_IGS	**1**	1	1	1	1	1	1
Development	ADV_FSP	**1**	1	1	**2**	**3**	3	**4**
	ADV_HLD		**1**	**2**	2	**3**	**4**	**5**
	ADV_IMP				**1**	**2**	**3**	3
	ADV_IMT					**1**	**2**	***3***
	ADV_LLD				**1**	1	**2**	2
	ADV_RCR	**1**	1	1	1	**2**	2	**3**
	ADV_SPM				**1**	**3**	3	3
Guidance Documents	AGD_ADM	**1**	1	1	1	1	1	1
	AGD_USR	**1**	1	1	1	1	1	1
Life Cycle Support	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR							
	ALC_LCD				**1**	**2**	2	**3**
	ALC_TAT				**1**	**2**	**3**	3
Tests	ATE_COV		**1**	**2**	2	2	**3**	3
	ATE_DPT			**1**	1	**2**	2	**3**
	ATE_FUN		**1**	1	1	1	**2**	2
	ATE_IND	**1**	**2**	2	2	2	2	**3**
Vulnerability Assessment	AVA_CCA					**1**	**2**	2
	AVA_MSU			**1**	**2**	2	**3**	3
	AVA_SOF		**1**	1	1	1	1	1
	AVA_VLA		**1**	1	**2**	**3**	**4**	4

Table 6: Evaluation assurance level summary

Evaluation assurance level 1 (EAL1) - functionally tested

“EAL1 is applicable where some confidence in correct operation is required, but the threats to security are not viewed as serious. It will be of value where independent assurance is required to support the contention that due care has been exercised with respect to the protection of personal or similar information.

EAL1 provides an evaluation of the TOE as made available to the customer, including independent testing against a specification, and an examination of the guidance documentation provided. It is intended that an EAL1 evaluation could be successfully conducted without assistance from the developer of the TOE, and for minimal outlay. An evaluation at this level should provide evidence that the TOE functions in a manner consistent with its documentation, and that it provides useful protection against identified threats.“

Evaluation assurance level 2 (EAL2) - structurally tested

“EAL2 requires the co-operation of the developer in terms of the delivery of design information and test results, but should not demand more effort on the part of the developer than is consistent with good commercial practice. As such it should not require a substantially increased investment of cost or time.

EAL2 is therefore applicable in those circumstances where developers or users require a low to moderate level of independently assured security in the absence of ready availability of the complete development record. Such a situation may arise when securing legacy systems, or where access to the developer may be limited.“

Evaluation assurance level 3 (EAL3) - methodically tested and checked

“EAL3 permits a conscientious developer to gain maximum assurance from positive security engineering at the design stage without substantial alteration of existing sound development practices.

EAL3 is applicable in those circumstances where developers or users require a moderate level of independently assured security, and require a thorough investigation of the TOE and its development without substantial re-engineering.“

Evaluation assurance level 4 (EAL4) - methodically designed, tested, and reviewed

“EAL4 permits a developer to gain maximum assurance from positive security engineering based on good commercial development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. EAL4 is the highest level at which it is likely to be economically feasible to retrofit to an existing product line.

EAL4 is therefore applicable in those circumstances where developers or users require a moderate to high level of independently assured security in conventional commodity TOEs and are prepared to incur additional security-specific engineering costs.“

Evaluation assurance level 5 (EAL5) - semiformally designed and tested

“EAL5 permits a developer to gain maximum assurance from security engineering based upon rigorous commercial development practices supported by moderate application of specialist security engineering techniques. Such a TOE will probably be designed and developed with the intent of achieving EAL5 assurance. It is likely that the additional costs attributable to the EAL5 requirements, relative to rigorous development without the application of specialised techniques, will not be large.

EAL5 is therefore applicable in those circumstances where developers or users require a high level of independently assured security in a planned development and require a rigorous development approach without incurring unreasonable costs attributable to specialist security engineering techniques.“

Evaluation assurance level 6 (EAL6) - semiformally verified design and tested

“EAL6 permits developers to gain high assurance from application of security engineering techniques to a rigorous development environment in order to produce a premium TOE for protecting high value assets against significant risks.

EAL6 is therefore applicable to the development of security TOEs for application in high risk situations where the value of the protected assets justifies the additional costs.“

Evaluation assurance level 7 (EAL7) - formally verified design and tested

“EAL7 is applicable to the development of security TOEs for application in extremely high risk situations and/or where the high value of the assets justifies the higher costs. Practical application of EAL7 is currently limited to TOEs with tightly focused security functionality that is amenable to extensive formal analysis.“

Strength of TOE security functions (AVA_SOF)

AVA_SOF Strength of TOE security functions

“Strength of function analysis addresses TOE security functions that are realised by a probabilistic or permutational mechanism (e.g. a password or hash function). Even if such functions cannot be bypassed, deactivated, or corrupted, it may still be possible to defeat them by direct attack. A level or a specific metric may be claimed for the strength of each of these functions. Strength of function analysis is performed to determine whether such functions meet or exceed the claim. For example, strength of function analysis of a password mechanism can demonstrate that the password function meets the strength claim by showing that the password space is sufficiently large.”

Three levels exist: SOF-basic, SOF-medium, and SOF-high.

Vulnerability analysis (AVA_VLA)

AVA_VLA Vulnerability analysis

"Vulnerability analysis consists of the identification of flaws potentially introduced in the different refinement steps of the development. It results in the definition of penetration tests through the collection of the necessary information concerning: (1) the completeness of the TSF (does the TSF counter all the postulated threats?) and (2) the dependencies between all security functions. These potential vulnerabilities are assessed through penetration testing to determine whether they could, in practice, be exploitable to compromise the security of the TOE."

Application notes

"A vulnerability analysis is performed by the developer in order to ascertain the presence of security vulnerabilities, and should consider at least the contents of all the TOE deliverables including the ST for the targeted evaluation assurance level. The developer is required to document the disposition of identified vulnerabilities to allow the evaluator to make use of that information if it is found useful as a support for the evaluator's independent vulnerability analysis.

The intent of the developer analysis is to confirm that no identified security vulnerabilities can be exploited in the intended environment for the TOE and that the TOE is resistant to obvious penetration attacks.

Obvious vulnerabilities are considered to be those that are open to exploitation that requires a minimum of understanding of the TOE, skill, technical sophistication, and resources. These might be suggested by the TSF interface description. Obvious vulnerabilities include those in the public domain, details of which should be known to a developer or available from an evaluation authority.

Performing a search for vulnerabilities in a systematic way requires that the developer identify those vulnerabilities in a structured and repeatable way, as opposed to identifying them in an ad-hoc fashion. The associated evidence that the search for vulnerabilities was systematic should include identification of all TOE documentation upon which the search for flaws was based.

Independent vulnerability analysis goes beyond the vulnerabilities identified by the developer. The main intent of the evaluator analysis is to determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low (for AVA_VLA.2), moderate (for AVA_VLA.3) or high (for AVA_VLA.4) attack potential. To accomplish this intent, the evaluator first assesses the exploitability of all identified vulnerabilities. This is accomplished by conducting penetration testing. The evaluator

should assume the role of an attacker with a low (for AVA_VLA.2), moderate (for AVA_VLA.3) or high (for AVA_VLA.4) attack potential when attempting to penetrate the TOE. Any exploitation of vulnerabilities by such an attacker should be considered by the evaluator to be “obvious penetration attacks” (with respect to the AVA_VLA.*.2C elements) in the context of the components AVA_VLA.2 through AVA_VLA.4.”

Part D

Security Target

Attached is the Security Target: "*DataSIGN(tm) Security Token, Version 1.0, EAL4 – SOF high, Security Target*"

Author: First Data Corporation

Date: 2005-11-17

Version: 1.5

DataSIGN(tm) Security Token, Version 1.0
EAL4 - SOF high
Security Target

Version: **1.5**

DocumentID: **ST_001.15**

Date: **2005-11-17**

Authors: Curt Beeson, Michael Hodges, First Data Corporation

Document History

Version	Date	Changes	Editor
0.1	2002-08-20	Initial Version.	FDC
0.2	2002-09-10	Initial version, amended by input provided by FDC in chapter: 5.1.1 TOE security functional requirements.	FDC
0.3	2002-09-12	Change to EAL4 and usage of 163 bit ECDSA key. Implemented amendments made by FDC.	FDC
0.4	2003-02-03	Approval by FDC and Certicom during workshop.	FDC
0.5	2003-02-20	Additions according TOE components for verification and approval by FDC and Certicom	FDC
0.6	2003-03-19	Editorial reworks after first comments of evaluation body	FDC
0.7	2003-04-04	Additional editorial reworks after first comments of evaluation body	FDC
0.8	2003-05-13	Additional editorial reworks after first comments of evaluation body and internal review	FDC
0.9	2003-05-27	Additional rework regarding the RNG, SMS library version and CM system	FDC
1.0	2003-06-06	Finalisation according comments of certification body	FDC
1.1	2003-10-29	Rework regarding the usage of SW RNG	FDC
1.2	2004-01-19	Additional rework regarding the usage of SW RNG (FCS_RND)	FDC
1.3	2004-11-25	Additional rework regarding comments /Infineon certificate	FDC
1.4	2005-08-18	Change to EAL4 plain	FDC
1.5	2005-11-17	Minor corrections according to certification body comments	FDC

Table of contents

1 ST introduction

1.1 ST identification

Document Title:	DataSIGN(tm) Security Token, Version 1.0, Security Target
Document ID:	ST_001.15
Document Version:	1.5
Date of Version:	2005-11-17
Origin:	First Data Secure LLC
TOE Reference:	DataSIGN(tm) Security Token, Version 1.0
TOE Commercial Name:	DataSIGN(tm) Security Token
TOE Short Description:	Silicon chip including firm- and software, which is to be integrated in SmartCards, USB-Dongles, etc., which performs an electronic signature function on data presented to it
Product Type:	Smart Card (typical, others possible: e.g. Dongle)
Evaluation Type:	Composite Firm-/Software and Hardware evaluation
Evaluation Body:	Evaluation Body of TUV Informationstechnik GmbH, Germany
Certification Body:	Certification Body of TUV Informationstechnik GmbH, Germany

This ST is based upon Common Criteria, Version 2.1 ([CC]).
The TOE consists of the following components:

Component	Hardware (HW) Firmware (FW) Software (SW)	Version	Supplier	Reference to TOE documentation
Security Controller SLE 66C42P	HW	m1495, a14 CertID.: TUViT-DSZ-CC-9243-2005 from 31.05.2005	Infineon	[ST_IC]
RMS Library	FW	RMS (version 0.8)	Infineon	[ST_IC]
ROM Mask	FW	STS (version 53.5E.12)	Infineon	[ST_IC]
DataSIGN(tm) Soft- &Firmware	SW	2.0.3.0	Certicom	[SC_2030]

Table 1: TOE components

The following parties are involved during DataSIGN(tm) Security Token development and production process:

First Data Corporation / First Data Secure LLC 10825 Farnam Dr. – C18 Omaha, NE 68154 USA	Initiator, product development
Certicom 5520 Explorer Drive, 4th Floor, Mississauga, Ontario, Canada L4W5L1	Software development DataSIGN(tm) Security Token
Infineon Technologies AG St.-Martin-Straße 76 81609 München Germany	Hardware development DataSIGN(tm) Security Token

1.2 ST overview

The main objectives of this Security Target are:

- To describe the Target of Evaluation (TOE). This ST focuses on the First Data Secure LLC silicon chip, which consists of hardware as well as of firm- and software. The hardware has already been evaluated in a separate evaluation. Therefore this evaluation is a composite Firm-/Software and Hardware evaluation.

  All issues related with the Infineon Security Controller SLE 66C42P have already been covered by the hardware evaluation CertID: TUViT-DSZ-CC-9243-2005.

- To describe the security environment of the TOE including the assets to be protected and the threats to be countered by the TOE and its environment.
- To describe the security objectives of the TOE and it's supporting environment.
- To specify the security requirements, which include the TOE security functional requirements as of CC, part 2 and the assurance requirements as of CC, part 3.
- To set up the TOE summary specification, which includes the TOE security functions specifications and the assurance measures.

1.3 ISO/IEC 15408 (CC) Conformance

This ST is claimed to be conformant with the ISO/IEC 15408:1999 (Common Criteria, Version 2.1, see [CC]) and its following parts

- Part 2 extended by FCS_RND.1 and
- Part 3 conformant

The minimum strength level of the TOE security functions is **SOF-high**.

The assurance level is **EAL 4.**

2 TOE description

2.1 Overview

First Data Corporation developed an online system supported by an authentication database, which will have the ability to monitor the complete life cycle of a security token beginning with the design and production of the ROM mask to the secure disposal of the token. Intention of this system is to connect every single security token production batch with a kind of security rating (score). The score will correspond to the trust a user can have in a signature performed by the token. The user might then decide to trust the signature e.g. for transactions up to a certain amount of money or a for certain valuable kind of contract etc.

In General the security level of a crypto product decreases during its life time. As long as the product is new its security functionality might be regarded to be very trustworthy, e.g. because it uses an up to date crypto algorithm with reasonable strong parameter settings (like key length). However, during life time of the crypto product, cryptological and other threats may be developed which would allow a threat agent (the bad guy) to alter the signed data. Altering data could be done before signing with the TOE without recognition of the card holder or after signing it might be done by recalculation of the signature. In both cases a valid signature would be presented to the verification authority of the receiver for the threat agent's own end. In order to provide the user of the product with a possibility to judge the trustworthiness of the crypto product, the DataSIGN(tm) online system has been developed. It basically consists of a security token and a database. The database contains the public key of any existing security token. Connected with the public key is the security rating (score) for the single security token. The score is also stored in the database. The score will range from 0 to 100. It will have a high rating (e.g. 80) as long as the security functionality of the token is proven to be resistant against any attack. During life time of the token however, the score will decrease as technology develops and the tokens security measures won't be able to resist up to date attacks any more. The score can reach 0 in case the token is regarded to be fully insecure.

Typical usage case of the security token and its background system (example):

Secure bank transaction

A bank customer (user) signs a bank transaction with his security token. The signed message contains his bank account number, the transaction data, the PIN status of the security token and the signature. The bank receives the signed data and wants to evaluate whether it can trust the signed message it received. In order to do this, the bank first identifies the user by his account number and checks the signed message with

the public key the user has provided when opening the account. If the crosscheck of the signature gives a positive result, the message has very probably been signed by this bank customer. However, the bank can not be sure that the message is authentic. It still might be possible that somebody has altered the signed message in order to perform an unauthorized transaction. This depends on the strength of the security functionality of the token on that particular day. The bank therefore sends a request to the DataSIGN(tm) database. The database will give back the security score of that particular token on that particular day. The score now allows the bank to decide on either to accept the transaction - in case the score is high enough – or to decline it – in case the score is low.

The overall objective is a security certification of the whole system. The first step however, is the evaluation and certification of the DataSIGN(tm) Security Token. The target of evaluation (TOE) therefore basically is the silicon chip (DataSIGN(tm) Security Token) which performs the signature functionality.

2.2 TOE definition

The Target of Evaluation has been indicated to be a silicon chip - the DataSIGN(tm) Security Token - which consists of the components listed in Table 1: TOE components in chapter 1.1 ST identification. Out of these the main components are:

a.) the Infineon smart card security controller SLE 66C42P (m1495, a14) - it includes all hard- and firmware related items as of Table 1: TOE components - and

b.) the DataSIGN(tm) embedded software which combines a smart card operating system and all required software functionality (includes software related items as of Table 1: TOE components).

The TOE is the combination of the main components a.) and b.). Whereas the main component a.) has already been evaluated according to CC EAL 5+ in a separate evaluation. During the combined evaluation of a.) and b.) all relevant findings of the previous evaluation EAL 5+ of a.) will be covered without performing a re- evaluation of a.).

The operating system including the application functionality is the SW component as listed in Table 1: TOE components. The libraries delivered by Infineon are considered to be firmware components.

2.3 Description of TOE security functionality

The DataSIGN(tm) Chip will provide an electronic signature on any data presented to it. The TOE automatically adds a PIN status data field to the data to be signed. The PIN status provides information for the recipient of the signed message about the fact, that

there was a valid PIN presented just before or a valid PIN was presented some signatures before or no PIN was presented any time before or the last PIN verification before signature had failed. In the cases of no valid PIN presented just before the recipient only can trust the messages integrity, not however, that it has been signed by a certain person. Furthermore, the TOE contains the optional feature of adding User Data (like a bank account number) to the data to be signed before signature. If User Data is stored on the TOE it is securly stored, that means either it can be changed only after succesful verification of the PIN or it is configured to be unchangeable. Depending on the configuration the TOE also is enabled or disabled to insert the User Data into the data to be signed. The User Data is stored as XML formated Tags, if User Data is allowed to be added only the appropriate data of the same XML objects will be replaced in the data to be signed.

The TOE will then provide a signature over the following set of data:

- Data to be signed, provided from the user
- PIN status data
- (optional) User data stored in the TOE

The DataSIGN(tm) Security Token uses SHA-1/ECDSA 163 bit in order to create the signature.

TOE Security Functionality Summary:

- Generation of a key pair for 163 bit K-163 ECDSA signature creation and export of the public key.
- Provides a K-163 ECDSA signature on data presented to it.
- PIN based user identification and authentication is required to prevent the TOE from adding bad PIN status information to the data to be signed before signature.
- User data stored in the TOE is added to the data to be signed before signature depending on the configuration of the TOE and the kind of XML objects as appropriate for replacement of signing data.
- If present User data is securely stored in the TOE.

2.4 TOE Configuration

The DataSIGN(tm) Security Token can only be configured once during the production process at the stage of power on testing. The configuration is restricted to the setting of the following variables:

- PIN (8 bytes)
  The PIN has to be a 8 Byte value. It can be alphanumerical. The PIN can be pre set to a certain value during configuration or set at a later time by the token holder.
- PIN block counter (1 byte)
  Presets the maximum tries before the DataSIGN(tm) Security Token blocks.
- Firmware configuration data (3 bytes)

  First byte (bit 1 is least significant and bit 8 is most significant):

  Bit 1 CHANGE PIN command. 0 for single run, 1 for multiple run.

  Bit 2 User data changeable. 0 for unchangeable, 1 for changeable.

  Bit 3 EXPORT PUBLIC KEY command. 0 for single run, 1 for multiple run.

  Bit 4 ECDSA SIGN command. 0 to disable special XML processing, 1 to enable.

  Bit 5 Random number generator. 0 for hardware based RNG, 1 for [AIS 20] conformant deterministic random generator (hardware seeding); 1 is required for bit 5

  Bit 6 Power On Self Test (POST). 0 to disable cryptographic POST, 1 to enable.

  Bit 7 Default PIN usage. 0 to treat it like the user PIN, 1 to disable the VERIFY command until the default PIN has been changed.

  Bit 8 Reserved for Future Use (RFU). Must be set to 0.

  2nd and 3rd Byte are reserved for future use.
- The second and third bytes are reserved for future use, and must be set to zero.
- PIN change once/multiple times
  PIN change by the user is allowed once or multiple times.
- User data changeable yes/no
  The user data stored in the TOE can or can not be changed by the user if user presents the valid PIN.
- Export public key once/multiple times
  The public key can be exported only once after configuration or multiple times.
- XML processing for ECDSA signature enable/disable
  If enabled the user data will be used for XML processing. The TOE may than for example automatically add the users account number in case a bank transaction is going to be signed. XML processing is the systematic search for XML data

fields within the data to be signed and replacement of them with appropriate XML fields contained in the stored User Data.

- Random Number Generator (RNG): Two configurations are possible:
  Using the hardware RNG provided by the Infineon chip – not to be used with the TOE.

  Using the [AIS20] conformant software RNG with hardware seeding – required.

- Power On Self Test enable/disable

  If enabled, the tests performed are:

  Cryptographic Algorithm Test – Two known answer test are performed: ECDSA signing (with a fixed key, a fixed random data stream and a fixed message), and SHA1 hash (fixed data).

  Software Integrity Test – A 16-bit CRC is used for checking the integrity of the software.

  Critical Functions Test – A memory test is performed to check for stuck bits of XRAM of the SLE66 processor.

- Default PIN change forced yes/no
  If set to yes, the user will be forced to enter a new PIN before he will be able to sign messages with positive PIN status information.

- RFU use. Must be set to 0.
  Reserved for future usage.

- Reserved (4 bytes), must be set to 0.
  Reserved for future usage.

After configuration during the first Power On Self Test (POST) at the production level, the TOE is regarded to be finished for delivery. After this setting, it is not possible to perform any security relevant changes to the TOE. However, for delivery to the end user some further production finalisation steps like packaging, etc. have to be performed. These steps are not regarded to be part of the TOE delivery process as the security relevant production of the TOE is complete after leaving the silicon manufacturers site.

The RNG is either configurable as a HW RNG or as SW RNG. Inside this composite evaluation the HW RNG is usable as seed for the SW RNG only, not as separated solution.

According to the Software based RNG the functionality class of the RNG is specified as K3 and the SoF high (see [AIS20]). This is related to SW RNG only, and the functional requirement FCS_RND.1 defines that the TSF of the software shall provide a mechanism to generate random numbers that meet the criteria specified in [AIS20].

2.5 TOE Development and Production

The product development has been performed by First Data Corporation. It has been transformed into the DataSIGN(tm) firm- and software by Certicom. The hardware, the Security Controller SLE 66C42P, is supplied by Infineon and has been evaluated according to **EAL 5+, SoF high** in a separate evaluation process.

Security during development and production therefore needs to be provided for this evaluation especially for the following processes:

- secure Firm- and Software development at Certicom.
- secure Firm- and Software transfer into the hardware platform by Certicom, First Data Corporation and Infineon.
- secure TOE configuration and testing at Infineon.

Hardware production related security requirements are already covered by the hardware evaluation of the Infineon chip.

2.6 TOE Life Cycle

The TOE life cycle can be illustrated like this:

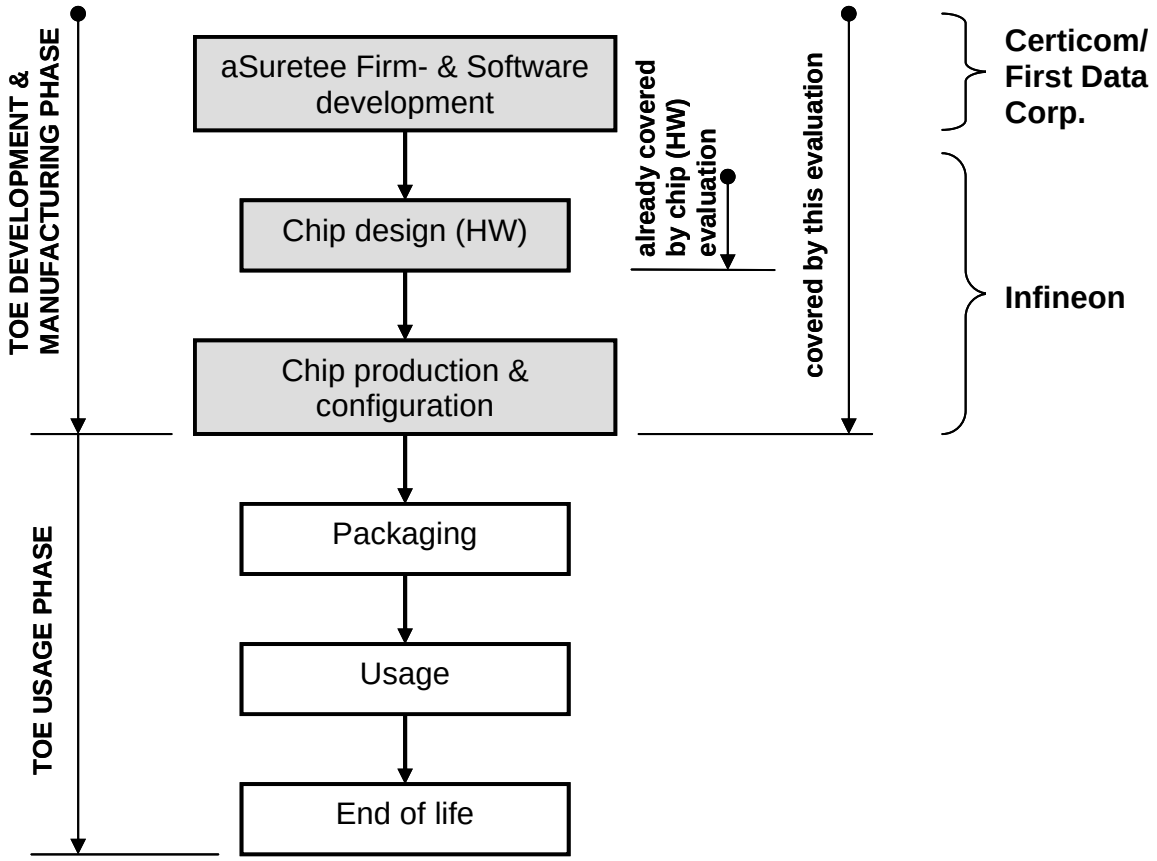


Figure 1 - TOE life cycle

2.6.1 TOE Hardware Development

The hardware development is located at Infineon, Munich, Germany. All aspects of the hardware development already have been covered by the EAL 5+ evaluation of the silicon chip Infineon SLE 66C42P (see Table 1: TOE components for TUVITDSZ-CC-9243-2005). The relevant findings of the previous hardware evaluation will be covered by the current evaluation of the DataSIGN(tm) chip.

2.6.2 TOE Firm- & Software Development

The TOE firm- & software development is located at Certicom, Mississauga, Ontario, Canada. All Firm- and Software development has been performed there. Thus, the Certicom site has taken all required security measures in order to satisfy the CC requirements.

The software quality control is performed by First Data Corp. in Omaha, Nebrasca, USA. Whereas no changes to the firm- & software are conducted during this process. In case a problem is detected during the control process at First Data Corporation, the information about the required changes will be handed over to Certicom. Certicom will then provide an firm- & software update to First Data Corporation for quality control purposes. As soon as the quality check has been performed successful, the firm- and software package will be handed over to Infineon for implementation in the chip during the production process. The firm- and software transfer from First Data Corporation to Infineon will use secure processes in order to make sure that the integrity of the firm- & software is preserved.

2.6.3 TOE Usage

In the TOE usage environment all security relevant settings (see also section TOE configuration) of the TOE are fixed and not changeable to a user or any other person. The TOE can be securely used in a non public environment which provides protection against PIN alteration and theft.

The TOE itself requires basic security relevant support from its usage environment. However, applications may be developed which are making use of the TOE and which require more sophisticated security relevant support from their environment. As these applications do not belong to the TOE and do not provide security functional support to the TOE, they and their possible functionality are not covered by this evaluation.

Please also see chapter “TOE definition” for further details about the TOE usage.

2.6.4 TOE End of Life

There is no special end of life environment or even end of life status specified for the TOE. Even in case the wrong PIN counter has reached its maximum value the TOE will go on performing electronic signatures. However, the PIN status information which will automatically be added to the data to be signed before signature will of course indicate that no valid PIN has been presented to the TOE before signature and additionally the token was zeroised before. That means that the signature will be calculated with a

zeroised key and therefore it will not be possible to produce valid signatures with the original private key any more.

2.6.5 Actors and Roles

The TOE knows the following actors and roles:

- Configurator: Person or device performing the TOE configuration during the Power On Self Test (POST) at Infineon.
- Owner Person or device handling the TOE in usage phase and has knowledge of the PIN.
- User: Person or device handling the TOE in usage phase.
- Other: All other persons or devices.

2.7 TOE Boundaries

2.7.1 Physical Boundaries

The complete surface of the chip has been chosen to be the boundary of the TOE to the outside world.

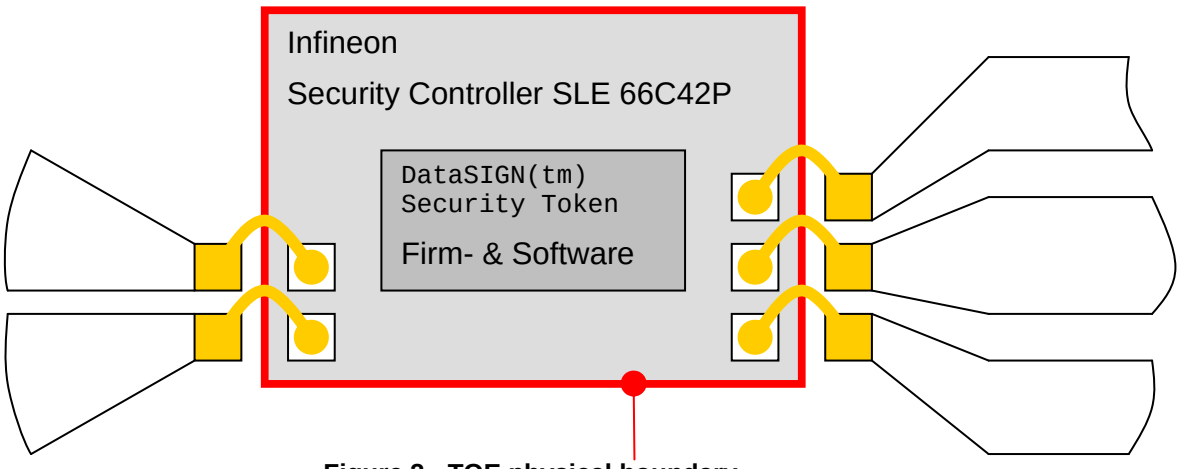


Figure 2 - TOE physical boundary

In order to become the DataSIGN(tm) Security Token product, packaging of any kind is required for the TOE (e.g. Chip card, USB Dongle, etc.). This means that the DataSIGN(tm) Security Token (silicon chip) which is the TOE needs to be mounted in any way to be able to get electrically connected to the outside world. Furthermore a kind

of coverage of the TOE is required in order to be mechanically protected from its environment. During this process no changes to the silicon chip (TOE) are performed.

2.7.2 TOE Logical Boundaries

The logical TOE boundary has been chosen to be the APDU command interface to the DataSIGN(tm) Security Token Firm- & Software. The DataSIGN(tm) product can be used from the outside world via its logical interface which equals the TOEs logical interface. Hence, the logical access and functionality of the product and the TOE are the same.

3 TOE security environment

3.1 Assets

The following assets have been identified:

Asset Name	Description
D.Signature (Hash function, signature algorithm)	**Digital signature** of data presented to the TOE for signature including PIN status information and additional user data stored in the TOE (e.g. account #, etc.) The digital signature is calculated of all the following items: - data presented to the TOE, - D.PINStatus, and - additional user data stored in the TOE in case of XML-processing is activated Integrity of the digital signature must be maintained.
D.PINStatus	**PIN Status** data field indicates if the correct PIN has been presented to the TOE before signature or not. The PIN status data will be added to the data to be signed before signature. The following four PIN stati are possible: - a valid PIN was presented just before the signature (0x00). - a valid PIN was presented one ore more signatures before (0x01). - no valid PIN was presented (0x02). - the last PIN verification failed (0x03). Integrity of the PIN status within the TOE must be maintained.
D.KeyPriv	**Private signature key** is an asset because an attacker (unauthorized key holder) would be able sign messages in the name of the authorized user. Integrity and confidentiality of the private signature key must be maintained.
D.KeyPub	**Public signature key** is an asset as it will be used to perform electronic signature verification. Integrity of the public signature key must be maintained when it is exported.

Asset Name	Description
D.IDData (PIN, User Data)	**Identification data: PIN and User Data** whereas the PIN is the users personal identification number which allows the user to change the User Data. User Data is a special individual user information which can be stored in the TOE during configuration (POST) and – depending on the TOEs configuration - might be changeable during the TOEs usage phase. Integrity of the User Data needs to be maintained. Confidentiality and authenticity of the PIN needs to be maintained.
D.ConfData	**Configuration Data** of the DataSIGN(tm) chip. The configuration data is being set once during Power On Self Test (POST) in production phase: o Initial PIN o PIN block counter (1-255 attempts) o PIN change allowed once/multiple times o User data changeable yes/no o Export public key once/multiple times o XML processing for ECDSA signature enable/disable o Random Number generator Hardware/Software with hardware seeding o Power On Self Test enable/disable o Default PIN change forced yes/no Integrity of configuration data needs to be maintained.

Table 2: Assets

3.2 Assumptions (about the environment)

The TOE handling during the following phases is security relevant:

- DataSIGN(tm) Firm- & Software development
- Chip design (HW)
- Chip production & configuration

These phases are pre usage phases. During the pre usage phases a special security supporting environment is required in order to manufacture and configure the TOE in a secure way. All the security requirements in the areas of chip design (HW) and DataSIGN(tm) Firm- & Software development will automatically be covered during the evaluation by CC assurance classes ACM, ADO and ALC as they are process related.

All TOE related security requirements in the phase chip production & configuration is going to be covered in the following sections.

After the 'Chip production & configuration' phase no security relevant changes can be performed on the TOE any more.

The following assumptions need to be made about the TOE environment:

Assumption Name	Description
A.Development&Production	Assumptions about the **TOE Development & Production environment** A secure production and configuration environment is assumed and the configurator is assumed to be trustworthy.
A.EnvUsage	Assumption about the **TOE usage environment** The usage environment is secure in a way that the TOE and the TOE communication data can not be accessed in an unauthorized way.
A.PINChange	Assumption about the **TOE PIN change procedure** If an institution does not want the end user to change the PIN, the PIN change option during configuration should be set to "change once". The institution should generate the PIN properly regarding length and alpha numerical characters, set it in the token, and communicate it to the end user using secure transmission. Therefore only the end user and the institution are assumed to know the PIN.
A.PINConfidentiality	Assumption about the **TOE PIN confidentiality** PIN is kept confidential by the user. The user therefore uses the TOE in a secure usage environment which enables the user to keep the PIN confidential, furthermore the user is informed about how to handle the PIN in a secure way.

Table 3: Assumptions

Remark: A.Development&Production – thrustworthy configurator

The Configurator is the person who is responsible for the correct configuration of the DataSIGN(tm) chip during Power On Self Test (POST) in the production phase. The Configurator can influence the following security features of the DataSIGN(tm) chip:

DataSIGN(tm) Configuration Data:

- Initial PIN
- PIN block counter (1-255 attempts)
- PIN change allowed once/multiple times

- User data changeable yes/no
- Export public key once/multiple times
- XML processing for ECDSA signature enable/disable
- Random Number generator Hardware/Software with hardware seeding
- Power On Self Test enable/disable
- Default PIN change forced yes/no

User Data:

- Content of UserData (this could be an account number)

The Configurator is explicitly allowed to change all of this data during production phase and therefore the Configurator has to be considered as trustworthy.

3.3 Threats

The following threat agents have been identified:

- User (Signer, Signature recipient)

  The threat agent User is regarded to be the ordinary user of the DataSIGN(tm) product with no more than usage related knowledge about the TOE and without knowledge of the PIN. The Users resources are limited to the equipment necessary in order to use the TOE. He can use generally available household and office tools to perform an attack. The time a User is willing to spend for an attack has to be regarded as limited to some hours. The Users motivation for an attack is the will to produce ‘valid signatures’ for content (data to be signed) which has been modified in any way. ‘Valid signatures’ in this case are signatures where the signed data contains a PINStatus field indicating that a valid PIN has been presented to the TOE before signature despite of the fact that no valid PIN has been presented.

  Furthermore the User could be interested in changing the so called UserData (this could be an account number) which is stored in the TOE.

  Due to EAL4 plain the User has low attack potential only.

- Owner (Signer, Signature recipient)

  The threat agent Owner is regarded to be the ordinary user of the DataSIGN(tm) product with no more than usage related knowledge about the TOE but with knowledge of the PIN. The Owners resources are limited to the equipment

necessary in order to use the TOE. He can use generally available household and office tools to perform an attack. The time an Owner is willing to spend for an attack has to be regarded as limited to some hours. The Owners motivation for an attack is the will to produce ‘valid signatures’ for content (data to be signed) which has been modified in any way. ‘Valid signatures’ in this case are signatures where the signed data contains a PINStatus field indicating that a valid PIN has been presented to the TOE before signature. The Owner could also be interested in changing the so called UserData (this could be an account number) which is stored in the TOE if he is not allowed to change by TOE configuration.

Due to EAL4 plain the Owner has low attack potential only.

- Other attacker

  The Other attacker can be any person different from the User. The Other attacker’s motivation can reach from a criminal up to a more ‘scientific’ approach for attacking the TOE. In any case the attacker’s goal is to be able to produce ‘valid signatures’ for content (data to be signed) which has been modified in any way. ‘Valid signatures’ in this case are signatures where the signed data contains a PIN Status field indicating that a valid PIN has been presented to the TOE before signature despite of the fact that no valid PIN has been presented.

  Furthermore the Other attacker could be interested in changing the so called UserData (this could be an account number) which is stored in the TOE.

  Due to EAL4 plain the other attacker has low attack potential only.

The following threats against the assets have been identified:

Asset	Threat	Threat description
D.Signature	T.SignatureAnalyse	The Other attacker could forge digital signatures using a crypto analytical and mathematical attack.
D.PINStatus	T.PINStatusChange	The User and Other attackers could manipulate the PIN status field using a direct hardware attack.
D.KeyPriv	T.KeyPrivReadOut	The Other attacker could read out the Private Key using a direct hardware attack.
	T.KeyPrivDPADFA	The Other attacker could retrieve the users Private Key by performing a DPA or DFA to the TOE.
	T.KeyPrivCalc	The Other attacker could recalculate the private key by using crypto analytical and mathematical means.
D.KeyPub	T.KeyPub	The Other attacker could alter the public key during export by accessing the data stream.
D.ConfData	T.ConfDataChange	The User, Owner or the Other attacker could alter the configuration data in contradiction to the configuration of the TOE.
D.IDData	T.IDDataGuess	The User or the Other attacker could guess the PIN by trying one after the other.
	T.IDDataPINCalc	The Other attacker could recalculate the PIN (valid only in case the PIN is not user changeable and has been generated during production phase)
	T.IDDataAlterate	The User, the Owner or the Other attacker could alter user data by accessing the TOE in an unauthorized way.

Table 4: Threats

3.4 Organisational security policies

It is not required for the TOE to follow an organisational security policy.

4 Security objectives

4.1 Security objectives for the TOE

The following security objectives have been defined for the TOE:

Objective Name	Objective Type	Description
O.TOECryptoOP	Preventative	The TOE shall provide cryptographic operations based upon a standard and state of the art algorithm including parameter settings (Key length, etc.) for creation of the signature key pair (D.KeyPriv and D.KeyPub) and calculation of the D.Signature.
O.TOEKeyProtect	Preventative	The TOE shall ensure that the private key is inaccessible other than for signing purposes
O.TOEAuthent	Preventative	The TOE ensures that users are uniquely authenticated in order to manage security attributes and user data
O.TOEHardw	Preventative and Detective	The TOE shall provide a secure hardware platform resistant against brute force hardware attacks. Furthermore the TOEs software functionality shall support the hardware measures
O.TOEPINblock	Preventative	The TOE shall provide a PIN block functionality which blocks the PIN Status after a reasonable secure amount of wrong attempts (e.g. after 3 attempts the PIN verification is blocked and the signature keys are destroyed by zeroization)

Table 5: Security objectives for the TOE

4.2 Security objectives for the environment

The following security objectives have been defined for the TOE environment:

Objective Name	Objective Type	Description
O.EnvDevProd	Preventative	The production and configuration environment needs to be secure in a way that the TOE and the TOE communication data can not be accessed in an unauthorized way. The Configurator is considered to be thrustworthy.

Objective Name	Objective Type	Description
O.EnvUsage	Preventative	The usage environment needs to be secure in a way that the TOE and the TOE communication data can not be accessed in an unauthorized way
O.EnvPINRnd	Preventative	If an institution does not want the end user to change the PIN, the PIN change option during configuration should be set to “change once”. The institution should generate the PIN properly, set it in the token, and communicate it to the end user.
O.EnvPINConf	Preventative	PIN is kept inaccessible and confidential by the user. The user therefore uses the TOE in a secure usage environment.

Table 6: Security objectives for the environment

5 Security requirements

5.1 TOE security requirements

5.1.1 TOE security functional requirements

The following functional requirements identify the TOE functional requirements. They have been drawn from the CC Part 2 functional requirements components, respectively independent defined (see FCS_RND1.1).

The following access control security functional policy (SFP) is being used as defined in this chapter. The SFP is based on the token states CONFIGURATOR, OWNER, USER and ERROR state. These states are not possible to exists all at the same time. ERROR state is possible for the whole life cycle, CONFIGURATOR state is only possible during production. USER and OWNER state are only possible after production. See also Figure 3. The production ends with the first POST.

The roles are corresponding to the token states, there is no CONFIGURATOR available after production is finished, e. g. then the CONFIGURATOR acts as an USER or as an OWNER if he has knowledge of the PIN.

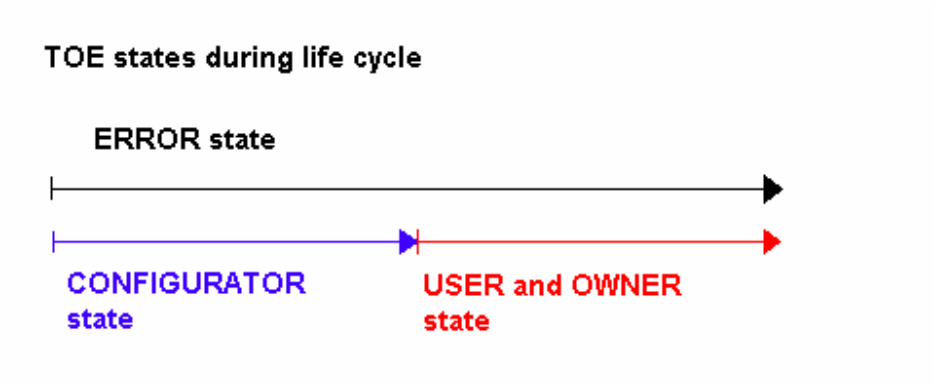


Figure 3 - Token states chronological order

Access Control SFP:

Object	Actions	Token State	Access
D.KeyPriv (Private Key)	NONE	CONFIGURATOR	*NA*
		USER	*NA*
		OWNER	*NA*
D.KeyPub (Public Key)	Export	CONFIGURATOR	Allowed
		USER	Allowed / Denied*
		OWNER	Allowed / Denied*
D.Signature	Sign	CONFIGURATOR	Allowed

Object	Actions	Token State	Access
(Digital Signature)		USER	Allowed
		OWNER	Allowed
D.IDData (User Data)	Export	CONFIGURATOR	Allowed
		USER	Allowed
		OWNER	Allowed
	Import	CONFIGURATOR	Allowed
		USER	Denied
		OWNER	Allowed / Denied*
D.IDData (PIN)	Export	CONFIGURATOR	Denied
		USER	Denied
		OWNER	Denied
	Import	CONFIGURATOR	Allowed
		USER	Denied
		OWNER	Allowed / Denied*
D.ConfData	Export except for PIN	CONFIGURATOR	Allowed
		USER	Allowed
		OWNER	Allowed
	Import	CONFIGURATOR	Allowed
		USER	Denied
		OWNER	Denied
D.PINStatus	Export	CONFIGURATOR	Allowed
		USER	Allowed
		OWNER	Allowed
	Import	CONFIGURATOR	Denied
		USER	Denied
		OWNER	Denied

*either Allowed or Denied according to configuration settings performed by the CONFIGURATOR during production.

Table 7: Access control SFP

Description of Token States:

CONFIGURATOR state is the configuration state of the token when power is applied the first time during production and the power on self test (POST) is performed.

USER state is the default state of the token and is the initialization state of the device at any power application after configuration has been completed. Basically, anyone who has physical access to the device after configuration.

OWNER state is an elevated operational state that is entered through the correct or validated completion of the token VERIFY command, any subsequent invalid VERIFY commands return the token to USER state. Removal of power returns the token to USER state. The Owner can be viewed as anyone who has physical access to the device as well as the PIN (authorized user).

Error state is the state when the token detects physical attacks or problems with hardware. In this state, all access of all subjects to objects is denied. The TOE is in **Error state** if it is in none of the states: **CONFIGURATOR, USER, and OWNER**.

Possible subjects according to the token states are listed in Table 8:

Token state	**Subjects**
CONFIGURATOR state	CONFIGURATOR
USER state	CONFIGURATOR, USER, OWNER
OWNER state	OWNER
Error state	CONFIGURATOR, USER, OWNER, OTHER

Table 8: Corresponding subjects to token states

The relation between objects and subjects from Table 7 has to be read as follows regarding Table 8 by replacement of the token states in Table 7 with the subjects in Table 8. For example the import of D.IDDATA (PIN) by the CONFIGURATOR, USER or OWNER during USER state has to be denied. The import of D.IDDATA (PIN) by the OWNER during OWNER state is either allowed or denied according to configuration settings performed by the CONFIGURATOR during production.

5.1.1.1 Class FAU: Security Audit

No audit functionality.

5.1.1.2 Class FCO: Communication

No communication functionality.

5.1.1.3 Class FCS: Cryptographic support

Family / Component / Element	Description	Dependencies
FCS_CKM.1	Cryptographic key generation requires cryptographic keys to be generated in accordance with a specified algorithm and key sizes which can be based on an assigned standard.	FCS_COP.1 FCS_CKM.4 FMT_MSA.2
FCS_CKM.1.1 ECDSA	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[ECDSA for curve K-163]** and specified cryptographic key sizes **[163 bit]** that meet the following: **[ANSI X9.62 and FIPS 186 standards]**.	
FCS_CKM.4	Cryptographic key destruction requires cryptographic keys to be destroyed in accordance with a specified destruction method which can be based on an assigned standard.	FCS_CKM.1 FMT_MSA.2
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **[Key zeroization]** that meets the following: **[FIPS140-2 standard]**.	
FCS_COP.1	Cryptographic operation requires a cryptographic operation to be performed in accordance with a specified algorithm and with a cryptographic key of specified sizes. The specified algorithm and cryptographic key sizes can be based on an assigned standard.	FCS_CKM.1 FCS_CKM.4 FMT_MSA.2

Family / Component / Element	Description	Dependencies
FCS_COP.1.1 ECDSA	The TSF shall perform **[ECDSA Sign with SHA-1 hashing and curve K-163]** in accordance with a specified cryptographic algorithm **[ECDSA]** and cryptographic key sizes **[of 163 bit]** that meet the following: **[ANSI X9.62, FIPS 186 standards]**.	
FCS_RND.1[1] FCS_RND.1.1	Quality metric for random numbers The TSF shall provide a mechanism to generate random numbers that meet **[the class K3 and SOF-high according to AIS 20].**	

Table 9: Class FCS

Remark: FCS_CKM.4.1 The TOE does not distinguish different users. It only knows one user, which can be regarded as both, authenticated and identified through presenting the correct PIN. As the TOE does not attempt to IDENTIFY different users, the purpose of the TOE is to provide a utilitarian digital signing capability that does not imply or endorse an identity but strictly digitally signs with a high level as it regard the actual signature not the signer. Furthermore the public key and the PIN are also zeroised when the defined number of unsuccessful authentication attempts has been met or surpassed (unsuccessful attempts >= PIN block counter).

5.1.1.4 Class FDP: User data protection

Family / Component / Element	Description	Dependencies
FDP_ACC.2	Complete access control requires that each identified access control SFP cover all operations on subjects and objects covered by that SFP. It further requires that all objects and operations with the TSC are covered by at least one identified access control SFP.	FDP_ACF.1

[1] This functional requirement is not taken from CC Part 2 (see chapter 8.2.1.1).

Family / Component / Element	Description	Dependencies
FDP_ACC.2.1 FDP_ACC.2.2	The TSF shall enforce the **[Access Control SFP]** on **[CONFIGURATOR, USER, OWNER, D.KeyPriv, D.KeyPub, D.Signature, D.PINStatus, D.IDData and D.ConfData]** and all operations among subjects and objects covered by the SFP.	
	The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.	
FDP_ACF.1	Security attribute based access control allows the TSF to enforce access based upon security attributes and named groups of attributes. Furthermore, the TSF may have the ability to explicitly authorise or deny access to an object based upon security attributes.	FDP_ACC.1 FMT_MSA.3
FDP_ACF.1.1	The TSF shall enforce the **[Access Control SFP]** to objects based on **[Token State (CONFIGURATOR/USER/OWNER)].**	
FDP_ACF.1.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[Operational State of the Token at the time of access attempt according to Access Control SFP (see Table 7 for more details)].**	
FDP_ACF.1.3	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[none]**.	
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[during Error State of the Token at the time of access attempt all access from all subjects to all objects is forbidden]**.	

Family / Component / Element	Description	Dependencies
FDP_ETC.1	Export of user data without security attributes requires that the TSF enforce the appropriate SFPs when exporting user data outside the TSF. User data that is exported by this function is exported without its associated security attributes.	FDP_ACC.1
FDP_ETC.1.1	The TSF shall enforce the **[Access Control SFP]** when exporting user data, controlled under the SFP(s), outside of the TSC.	
FDP_ETC.1.2	The TSF shall export the user data without the user data's associated security attributes.	
FDP_ITC.1	Import of user data without security attributes associated with the user data imported from outside the TSC.	FDP_ACC.1 FMT_MSA.3
FDP_ITC.1.1	The TSF shall enforce the **[Access Control SFP]** when importing user data, controlled under the SFP, from outside of the TSC.	
FDP_ITC.1.2	The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC.	
FDP_ITC.1.3	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: **[none].**	

Table 10: Class FDP

5.1.1.5 Class FIA: Identification & authentification

Family / Component / Element	Description	Dependencies
FIA_AFL.1	Requires that the TSF be able to terminate the session establishment process after a specified number of unsuccessful user authentication attempts. It also requires that, after termination of the session establishment process, the TSF be able to disable the user account or the point of entry (e.g. workstation) from which the attempts were made until an administrator-defined condition occurs.	FIA_UAU.1
FIA_AFL.1.1	The TSF shall detect when **[Selectable (1-255)]** unsuccessful authentication attempts occur related to **[VERIFY].**	
FIA_AFL.1.2	When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **[perform token zeroization]**.	
FIA_UAU.1	Timing of authentication, allows a user to perform certain actions prior to the authentication of the user's identity.	FIA_UID.1
FIA_UAU.1.1	The TSF shall allow **[Import of configuration data before POST, Export of configuration data except PIN and D.IDData (User Data) as well as the function Digital Signature Sign after first POST]** on behalf of the user to be performed before the user is authenticated.	
FIA_UAU.1.2	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.	
FIA_UAU.7	Protected authentication feedback, require that only limited feedback information is provided to the user during the authentication.	FIA_UAU.1

Family / Component / Element	Description	Dependencies
FIA_UAU.7.1	The TSF shall provide only **[information about Invalid Password Length]** to the user while the authentication is in progress.	
FIA_UID.1	Timing of identification, allows a user to perform certain actions prior to the authentication of the user's identity.	None
FIA_UID.1.1	The TSF shall allow **[Export of D.IDData (User Data) as well as the function Digital Signature Sign]** on behalf of the user to be performed before the user is identified.	
FIA_UID.1.2	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.	

Table 11: Class FIA

Remark: FIA_UAU.1 The export of Software/firmware version, Token status, Serial number of the chip and Elliptic curve supported after first POST on behalf of the user to be performed before the user is authenticated is also allowed but these data are not security relevant and therefore not defined as assets.

Remark: FIA_UID.1 The TOE does not distinguish different users. It only knows one user, which can be regarded as both, authenticated and identified through presenting the correct PIN. As the TOE does not attempt to IDENTIFY different users, the purpose of the TOE is to provide a utilitarian digital signing capability that does not imply or endorse an identity but strictly digitally signs with a high level as it regard the actual signature not the signer.

5.1.1.6 Class FMT: Security management

Family / Component / Element	Description	Dependencies
FMT_MSA.1	Management of security attributes allows authorised users (roles) to manage the specified security attributes.	FDP_ACC.1 FMT_SMF.1 FMT_SMR.1
FMT_MSA.1.1	The TSF shall enforce the **[Access control SFP]** to restrict the ability to **[*selection*: change_default or modify]** the security attributes **[PIN]** to **[OWNER].**	
FMT_MSA.2	Secure security attributes ensures that values assigned to security attributes are valid with respect to the secure state.	ADV_SPM.1 FDP_ACC.1 FMT_MSA.1 FMT_MSR.1
FMT_MSA.2.1	The TSF shall ensure that only secure values are accepted for security attributes.	
FMT_MSA.3	Static attribute initialisation ensures that the default values of security attributes are appropriately either permissive or restrictive in nature.	FMT_MSA.1 FMT_SMR.1
FMT_MSA.3.1	The TSF shall enforce the **[Access control SFP]** to provide **[*selection*: restrictive and permissive]** default values for security attributes that are used to enforce the SFP.	
FMT_MSA.3.2	The TSF shall allow the **[OWNER]** to specify alternative initial values to override the default values when an object or information is created.	
FMT_SMF.1	Specification of Management Functions	None
FMT_SMF.1.1	The TSF shall be capable of performing the following security management functions: **[Change PIN, Import User Data].**	
FMT_SMR.1	Security roles	FIA_UID.1
FMT_SMR.1.1	The TSF shall maintain the roles **[CONFIGURATOR, USER and OWNER].**	
FMT_SMR.1.2	The TSF shall be able to associate users with roles.	

Table 12: Class FMT

5.1.1.7 Class FPR: Privacy

No privacy functionality.

5.1.1.8 Class FPT: Protection of TOE security functions

Family / Component / Element	Description	Dependencies
FPT_AMT.1	Abstract machine testing	None
FPT_AMT.1.1	The TSF shall run a suite of tests **[*selection*: during initial start-up; *refinement:* of initial start-up during power-up (POST)]** to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.	
FPT_FLS.1	Failure with preservation of secure state	ADV_SPM.1
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur: **[over and under voltage, over and under clock frequency, temperature, electro magnetic radiation, physical probing, and firmware integrity problems].**	
FPT_TST.1	TSF testing provides the ability to test the TSF's correct operation. These tests may be performed at start-up, periodically, at the request of the authorised user, or when other conditions are met. It also provides the ability to verify the integrity of TSF data and executable code.	FPT_AMT.1
FPT_TST.1.1	The TSF shall run a suite of self tests **[*selection*: during initial start-up; *refinement* of initial start-up:during power-up (POST)]** to demonstrate the correct operation of the TSF.	
FPT_TST.1.2	The TSF shall provide authorised users with the capability to verify the integrity of TSF data.	

Family / Component / Element	Description	Dependencies
FPT_TST.1.3	The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.	

Table 13: Class FPT

5.1.1.9 Class FRU: Resource utilisation

No resource utilisation functionality.

5.1.1.10 Class FTA: TOE access

No TOE access functionality.

5.1.1.11 Class FTP: Trusted path / channels

No TOE trusted path/channel functionality.

5.1.2 TOE security assurance requirements

The Evaluation Assurance Level chosen for his Evaluation is **EAL 4**.

The following TOE assurance requirements drawn from CC Part 3 are valid:

Identification	Description	Direct dependencies
ACM	**Configuration management (CM)**	
ACM_AUT.1	Partial CM automation	ACM_CAP.3
ACM_CAP.4	CM Capabilities	ACM_SCP.1 ALC_DVS.1
ACM_SCP.2	CM Scope	ACM_CAP.3
ADO	**Delivery and Operation**	
ADO_DEL.2	Delivery	ACM_CAP.3
ADO_IGS.1	Installation, generation and start-up procedures	AGD_ADM.1
ADV	**Development**	
ADV_FSP.2	Functional specification	ADV_RCR.1
ADV_HLD.2	Security enforcing high-level design	ADV_FSP.1 ADV_RCR.1
ADV_IMP.1	Implementation representation: Subset of the implementation of the TSF	ADV_LLD.1 ADV_RCR.1 ALC_TAT.1
ADV_LLD.1	Descriptive low-level design	ADV_HLD.2 ADV_RCR.1
ADV_RCR.1	Representation correspondence: Informal correspondence demonstration	None
ADV_SPM.1	Security policy model: Formal TOE security policy model	ADV_FSP.1
AGD	**Guidance documents**	
AGD_ADM.1	Administrator guidance	ADV_FSP.1
AGD_USR.1	User guidance	ADV_FSP.1
ALC	**Life cycle support**	
ALC_DVS.1	Development security: Identification of security measures	None

Identification	Description	Direct dependencies
ALC_LCD.1	Life cycle definition: Developer defined life-cycle model	None
ALC_TAT.1	Tools and techniques: Well-defined development tools	ADV_IMP.1
ATE	**Tests**	
ATE_COV.2	Coverage: Analysis of coverage	ADV_FSP.1 ATE_FUN.1
ATE_DPT.1	Depth of testing Testing: low–level design	ADV_HLD.1 ATE_FUN.1
ATE_FUN.1	Functional testing	None
ATE_IND.2	Independent testing	ADV_FSP.1 AGD_ADM.1 AGD_USR.1 ATE_FUN.1
AVA	**Vulnerability assessment**	
AVA_MSU.2	Validation of analysis	ADO_IGS.1 ADV_FSP.1 AGD_ADM.1 AGD_USR.1
AVA_SOF.1	Strength of TOE security function evaluation SoF high	ADV_FSP.1 ADV_HLD.1
AVA_VLA.2	Independent vulnerability analysis	ADV_FSP.1 ADV_HLD.2 ADV_IMP.1 ADV_LLD.1 AGD_USR.1

Table 14: TOE assurance requirements

5.1.3 Minimum strength of function (SoF) claim

The minimum strength of function claimed for this evaluation is **SoF high** (details see section 6.2).

The strength of cryptographic algorithms is outside the scope of the CC. Strength of function only applies to probabilistic or permutational mechanisms that are non-cryptographic. Therefore, the assessment of algorithmic strength does not form part of this evaluation.

5.2 Security requirements for the IT environment

There are no security requirements for the IT environment.

5.3 Security requirements for the non IT environment

The following security requirements exist for the non IT environment:

Requirement Name	Description
R.EnvDevProd	Requirement of a secure production process The production and configuration environment needs to be secure in a way that the TOE and the TOE communication data can not be accessed in an unauthorized way. Appropriate physical (building), organisational and personal measures need to be in place.
R.EnvUsage	Requirement of a secure usage environment The usage environment needs to be secure in a way that the TOE and the TOE communication data can not be accessed in an unauthorized way
R.EnvPINConf	Requirement of keeping the PIN secret PIN is kept inaccessible and confidential by the user. The user needs to be informed in clear language about this requirement as well as about the consequences of a lost or stolen PIN.
R.PINChange	Requirement of true randomized PIN If an institution does not want the end user to change the PIN, the PIN change option during configuration should be set to “change once”. The institution should generate the PIN properly, set it in the token, and communicate it to the end user.

Table 15: Security requirements for the non IT environment

6 TOE summary specification

6.1 Security functions

The following Security Functions have been determined:

TSF TOE Security Function	Description
SF_KeyGenerate	SF_KeyGenerate provides public/private key generation based on a standard and state of the art elliptic curves algorithm ([ECDSA for curve K-163]). The size of the generated keys will be 163 bit to meet ANSI X9.62 and FIPS 186 standards. The key pair generation will be initiated during production process. SF_KeyGenerate can only be performed once during the power on self test (POST) in production phase.
SF_UserAuthentication	SF_UserAuthentication is accomplished by having the user entering 8 bytes PIN which is being validated against reference authentication data which has been previously established. Depending on the TOE configuration the user is allowed to change the PIN or not. If PIN change is possible the user needs to enter the correct 8 bytes PIN before PIN change is possible. PIN block functionality is part of SF_UserAuthentication in order to ensure that the PIN related TOE functionality is being blocked after a reasonable number of wrong attempts (e.g. three attempts). The number of allowed attempts is to be set during configuration in production phase (the number of configurable possible attempts is limited to 255).
SF_SignatureCreation	SF_SignatureCreation provides a signature over supplied data, the current TOE PIN state (possible data suffix: 0x00, 0x01, 0x02, 0x03) and if configured also over the so called User Data (e.g. an account number). A standard and state of the art elliptic curves crypto algorithm (ECDSA Sign with SHA-1 hashing and cryptographic key sizes of 163 bit that meet the ANSI X9.62, FIPS 186 standards) will be used by the TOE. The signature will be provided regardless of actual authentication state.
SF_DataAccess	SF_DataAccess provides the ability to import/export User Data, Public Key and TOE Status depending upon the token state and the configuration settings done during production. Depending on the configuration of the TOE (see section 2.4) the rules from Table 7 are enforced. During Error state no access between subjects and objects is allowed.

TSF TOE Security Function	Description
SF_AssetProtection	SF_AssetProtection zeroizes the private key, public key and the PIN according to cryptographic key destruction method that meets the following FIPS140-2 standard. When the defined number of unsuccessful authentication attempts has been met or surpassed. If the TOE determines a physical attack (e.g. micro probing attacks), access to the TOE is denied until the next power up.

Table 16: Security Functions

Furthermore the Security Functions from the HW, the SLE66C42P, are part of the composite product. The functions are:

SEF1: Operating state checking

SEF2: Phase management with test mode lock-out

SEF3: Protection against snooping

SEF4: Data encryption and data disguising

SEF5: Random number generation

SEF6: TSF self test

SEF7: Notification of physical attack

SEF8: Memory Management Unit (MMU)

SEF9: Cryptographic support

These functions are explained in detail within [ST_IC] and their relation to the SF listed above will be pointed out in the functional specification.

6.2 Strength of function claims

The Evaluation Assurance Level chosen for this Evaluation is **EAL 4**.

The minimum strength of function claimed for this evaluation is **SoF high** for SF_UserAuthentication.

SF_KeyGenerate and SF_SignatureCreation use [AIS20] class K3 and SOF-high conform random numbers to generate keys and signatures.

6.3 Assurance measures

6.3.1 AM_ACM: CONFIGURATION MANAGEMENT

This assurance measure ensures the configuration management. The CM responsible is in charge to write the CM plan, use the CM system and validate the CM system in order to confirm that ACM_AUT.1, ACM_CAP.4 and ACM_SCP.2 components are completed.

During the DataSIGN(tm) development process the following CM tools have been used:

CVS: CVS uses a server/client architecture. The CVS Server is version 1.11.5. The Client is WinCVS version 1.0.6 which uses CVS Client 1.10.5.

6.3.2 AM_ADO: DELIVERY AND OPERATION

This assurance measure ensures the delivery and operation. The delivery responsible is in charge to write delivery documentation and validate it in order to confirm that the procedure is applied.

6.3.3 AM_ADV: DEVELOPMENT

This assurance measure ensures the development. The development responsible is in charge to design the TOE, write development documentation and validate it in order to confirm that the related security functional requirements are completed by security functions.

In order to satisfy the requirements of EAL 4 an informal security policy model can be used.

6.3.4 AM_AGD: GUIDANCE DOCUMENTS

This assurance measure ensures the guidance documents. The guidance responsible is in charge to write administrator and user guidance. The documentation provides the rules to use and administrate the TOE in a secured manner.

6.3.5 AM_ALC: LIFE CYCLE

This assurance measure ensures the life cycle. Life cycle responsible is in charge to confirm that the life cycle process is applied.

6.3.6 AM_ATE: TESTS

This assurance measure ensures the tests. The test responsible is in charge to write tests and execute it in order to confirm that the security functions are tested.

6.3.7 AM_AVA: VULNERABILITY ASSESSMENT

This assurance measure ensures the vulnerability assessment. The security responsible is in charge to confirm that the security measures are suitable to meet the TOE security objectives conducing a vulnerability analysis.

7 PP claims

There are no PP claims.

8 Rationale

8.1 Security objectives rationale

The following table shows the connection between Assets, Threats, TOE Security Objectives and TOE Security Functions and Requirements to the environment which are assumed to be fulfilled:

Asset	Threat	Security Objective	Security Function / Requirements
D.Signature (Hash function, signature algorithm)	T.SignatureAnalyse The Other attacker could forge digital signatures using a crypto analytical and mathematical attack	O.TOECryptoOP The TOE shall provide cryptographic operations based upon a standard and state of the art algorithm including parameter settings (Key length, etc.) for creation of the signature key pair (D.KeyPriv and D.KeyPub) and calculation of the D.Signature. This will prevent an attacker from succeeding with a crypto analytical or mathematical attack.	SF_KeyGenerate SF_SignatureCreation
D.PINStatus	T.PINStatusChange The User and Other attackers could manipulate the PIN status field using a direct hardware attack	O.TOEHardw The TOE shall provide a secure hardware platform resistant against brute force hardware attacks. Furthermore the TOEs software functionality shall support the hardware measures in order to prevent direct hardware attacks	SF_AssetProtection
D.KeyPriv	T.KeyPrivReadOut A Other attacker could read out the Private Key using a direct hardware attack	O.TOEHardw The TOE shall provide a secure hardware platform resistant against brute force hardware attacks. Furthermore the TOEs software functionality shall support the hardware measures in order to prevent direct hardware attacks	SF_AssetProtection

Asset	Threat	Security Objective	Security Function / Requirements
	T.KeyPrivDPADFA The Other attacker could retrieve the users Private Key by performing a DPA or DFA to the TOE	O.TOEKeyProtect The TOE shall ensure that the private key is inaccessible other than for signing purposes O.TOEHardw The TOE shall provide a secure hardware platform resistant against brute force hardware attacks. Furthermore the TOEs software functionality shall support the hardware measures in order to prevent direct hardware attacks	SF_KeyGenerate SF_SignatureCreation SF_AssetProtection
	T.KeyPrivCalc The Other attacker could recalculate the key by using crypto analytical and mathematical means	O.TOECryptoOP The TOE shall provide cryptographic operations based upon a standard and state of the art algorithm including parameter settings (Key length, etc.) for creation of the signature key pair (D.KeyPriv and D.KeyPub) and calculation of the D.Signature. This will prevent an attacker from succeeding with a crypto analytical or mathematical attack.	SF_KeyGenerate SF_SignatureCreation
D.KeyPub	T.KeyPub The Other attacker could alter the public key during export by accessing the data stream	O.EnvDevProd The production and configuration environment needs to be secure in a way that the TOE can not be accessed in an unauthorized way. The configurator is considered to be thrustworthy. O.EnvUsage The usage environment needs to be secure in a way that the TOE and the TOE communication data can not be accessed in an unauthorized way	To be covered by the TOE environment. R.EnvDevProd R.EnvUsage

Asset	Threat	Security Objective	Security Function / Requirements
D.IDData (PIN, user data)	T.IDDataGuess The User or the Other attacker could guess the PIN by trying one after The Other	O.TOEPINblock The TOE shall provide a PIN block functionality which blocks the PIN Status after a reasonable secure amount of wrong attempts (e.g. after 3 attempts the PIN verification is blocked and the signature keys are destroyed by zeroization)	SF_UserAuthentication
	T.IDDataPINCalc The Other attacker could recalculate the PIN (valid only in case the PIN is not user changeable and has been generated during production phase).	O.EnvPINRnd If an institution does not want the end user to change the PIN, the PIN change option during configuration should be set to “change once”. The institution should generate the PIN properly, set it in the token, and communicate it to the end user. O.EnvPINConf PIN is kept inaccessible and confidential by the user. The user therefore uses the TOE in a secure usage environment.	To be covered by the TOE environment. R.PINChange R.EnvPINConf R.EnvDevProd
	T.IDDataAlterate The User, the Owner or the Other attacker could alter user data by accessing the TOE in an unauthorized way.	O.TOEAuthent The TOE ensures that users are uniquely authenticated in order to manage security attributes and user data	SF_UserAuthentication SF_DataAccess
D.ConfData	T.ConfDataChange The User, the Owner or the Other attacker could alter the configuration data in contradiction to the configuration of the TOE.	O.TOEAuthent The TOE ensures that users are uniquely authenticated in order to manage security attributes and user data	SF_UserAuthentication SF_DataAccess

Table 17: Security objectives rationale

8.1.1 Assets coverage

Threats Assumptions/Assets	D.Signature	D.PINStatus	D.KeyPriv	D.KeyPub	D.IDData	D.ConfData
T.SignatureAnalyse	**X**					
T.PINStatusChange		**X**				
T.KeyPrivReadOut			**X**			
T.KeyPrivDPADFA			**X**			
T.KeyPrivCalc			**X**			
T.KeyPub				**X**		
T.IDDataGuess					**X**	
T.IDDataPINCalc					**X**	
T.IDDataAlterate					**X**	
T.ConfDataChange						**X**
A.Development&Production				**X**	**X**	
A.EnvUsage				**X**		
A.PINChange					**X**	
A.PINConfidentiality					**X**	

Table 18: Threats coverage

8.1.2 Security Objectives coverage

Threats, Assumptions / Security Objectives	O.TOECryptoOP	O.TOEKeyProtect	O.TOEAuthent	O.TOEHardw	O.TOEPINblock	O.EnvDevProd	O.EnvUsage	O.EnvPINRnd	O.EnvPINConf
T.SignatureAnalyse	**X**								
T.PINStatusChange				**X**					
T.KeyPrivReadOut				**X**					

Threats, Assumptions / Security Objectives	O.TOECryptoOP	O.TOEKeyProtect	O.TOEAuthent	O.TOEHardw	O.TOEPINblock	O.EnvDevProd	O.EnvUsage	O.EnvPINRnd	O.EnvPINConf
T.KeyPrivDPADFA		**X**		**X**					
T.KeyPrivCalc	**X**								
T.KeyPub						**X**	**X**		
T.IDDataGuess					**X**				
T.IDDataPINCalc								**X**	**X**
T.IDDataAlterate			**X**						
T.ConfDataChange			**X**						
A.Development&Production						**X**			
A.EnvUsage							**X**		
A.PINChange								**X**	
A.PINConfidentiality									**X**

Table 19: Security Objectives coverage

By providing cryptographic operations based upon a standard and state of the art algorithm including parameter settings (O.TOECryptoOP) the Other attacker is not able to recalculate the private signature key by using crypto analytical and mathematical means and he is not able to forge digital signatures using a crypto analytical and mathematical attack.

By ensuring that the private key is inaccessible other than for signing purposes (O.TOEKeyProtect) the Other attacker is unable to retrieve the users Private Key by performing a DPA or DFA to the TOE.

By ensuring that users are uniquely authenticated in order to manage security attributes and user data (O.TOEAuthent), the User, the Owner or the other attacker is unable to alter user data and configuration data by accessing the TOE in an unauthorized way.

By providing a secure hardware platform resistant against brute force hardware attacks with hardware security measures supported by the TOEs software functionality (O.TOEHardw) the Other attacker is unable to retrieve the users Private Key by performing a DPA or DFA to the TOE and he is also unable to read out the Private Key using a direct hardware attack. The User and Other attackers are unable to manipulate the PIN status field using a direct hardware attack.

By providing a PIN block functionality which blocks the PIN Status after a reasonable secure amount of wrong attempts (O.TOEPINblock) the User or Other attacker is likely unable to could guess the PIN by trying one after the other.

By the production and configuration environment which needs to be secure in a way that the TOE and the TOE communication data can not be accessed in an unauthorized way (O.EnvDevProd) the Other attacker is unable to alter the public key during export by accessing the data stream.

By the usage environment which needs to be secure in a way that the TOE and the TOE communication data can not be accessed in an unauthorized way (O.EnvUsage) the Other attacker is unable to alter the public key during export by accessing the data stream.

If an institution does not want the end user to change the PIN, the PIN change option during configuration should be set to “change once”. By generating the PIN properly, setting it in the token and communicating it to the end user (O.EnvPINRnd) the Other attacker is unable to recalculate the PIN (valid only in case the PIN is not user changeable and has been generated during production phase).

Keeping the PIN inaccessible and confidential by the user (Owner) with usage of the TOE in a secure usage environment (O.EnvPINConf) the Other attacker is unable to recalculate the PIN.

8.2 Security requirements rationale

8.2.1 Choice of TOE security functional requirements

The choice of functional requirements is based on the analysis of the security objectives for the TOE. The security objectives for the environment are complementary with the security objectives for the TOE and are technologically necessary to complete the objectives for the TOE.

8.2.1.1 Definition of the Family FCS_RND

To define the IT security functional requirements of the TOE an additional family (FCS_RND) of the class FCS (Cryptographic support) is defined here. This family describes the functional requirements for random generation.

FCS_RND Generation of random numbers

Family behaviour: This family defines quality requirements for the generation of random numbers, which are intended to be use for cryptographic purposes.

Component levelling: FCS_RND Generation of random numbers - 1

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

Management: FCS_RND.1 - There are no management activities foreseen.

Audit: FCS_RND.1 - There are no actions defined to be auditable.

FCS_RND.1 Quality metric for random numbers

Hierarchical to: No other components.

FCS_RND.1.1 The TSF shall provide a mechanism to generate random numbers that meet [assignment: a defined quality metric].

Dependencies: No dependencies.

Security Objective	Security Function	Security functional requirement (SFR)
O.EnvDevProd The production and configuration environment needs to be secure in a way that the TOE can not be accessed in an unauthorized way. The configurator is considered to be thrustworthy.	To be covered by the TOE environment. R.EnvDevProd	None
O.EnvUsage The usage environment needs to be secure in a way that the TOE and the TOE communication data can not be accessed in an unauthorized way	To be covered by the TOE environment. R.EnvUsage	None
O.EnvPINConf PIN is kept inaccessible and confidential by the user. The user therefore uses the TOE in a secure usage environment.	To be covered by the TOE environment. R.PINConfidentiality	None
O.EnvPINRnd If an institution does not want the end user to change the PIN, the PIN change option during configuration should be set to “change once”. The institution should generate the PIN properly, set it in the token, and communicate it to the end user.	To be covered by the TOE environment. R.PINChange	None
O.TOEAuthent The TOE ensures that users are uniquely authenticated in order to manage security attributes and user data.	SF_UserAuthentication SF_DataAccess	FDP_ACF.1 FIA_AFL.1 FIA_UAU.1 FIA_UAU.7 FIA_UID.1 FMT_MSA.1 FMT_MSA.2 FMT_MSA.3 FMT_SMF.1 FMT_SMR.1 FDP_ACC.2 FDP_ETC.1 FDP_ITC.1

Security Objective	Security Function	Security functional requirement (SFR)
O.TOECryptoOP The TOE shall provide cryptographic operations based upon a standard and state of the art algorithm including parameter settings (Key length etc.) for creation of the signature key pair (D.KeyPriv and D.KeyPub) and calculation of the D.Signature.	SF_KeyGenerate SF_SignatureCreation	FCS_CKM.1 FPT_AMT.1 FPT_TST.1 FCS_RND.1 FCS_COP.1 FCS_RND.1
O.TOEHardw The TOE shall provide a secure hardware platform resistant against brute force hardware attacks. Furthermore the TOEs software functionality shall support the hardware measures.	SF_AssetProtection	FCS_CKM.4 FIA_AFL.1 FPT_AMT.1 FPT_FLS.1 FPT_TST.1 (PowerOnTest FIPS 140)
O.TOEKeyProtect The TOE shall ensure that the private key is inaccessible other than for signing purposes	SF_KeyGenerate SF_SignatureCreation SF_DataAccess	FCS_CKM.1 FPT_AMT.1 FPT_TST.1 FCS_RND.1 FCS_COP.1 FCS_RND.1 FDP_ACC.2 FDP_ETC.1 FDP_ITC.1
O.TOEPINblock The TOE shall provide a PIN block functionality which blocks the PIN Status after a reasonable secure amount of wrong attempts (e.g. after 3 attempts the PIN verification is blocked and the signature keys are destroyed by zeroization)	SF_UserAuthentication	FDP_ACF.1 FIA_AFL.1 FIA_UAU.1 FIA_UAU.7 FIA_UID.1 FMT_MSA.1 FMT_MSA.2 FMT_MSA.3 FMT_SMF.1 FMT_SMR.1

Table 20: Choice of TOE security functional requirements

8.2.1.2 Justification for suitability of SFR – TOE security objectives

Within the table above the complete constituent SFR of the SF are listed in the corresponding column of each row. For each TOE security objective it is justified that the TOE security requirements are suitable to meet that security objective as follows:

O.TOEAuthent

The TOE ensures that users are uniquely authenticated in order to manage security attributes and user data. This is realized by:

- FDP_ACF.1 because the TSF enforces the Access Control SFP based on Token States (CONFIGURATOR/USER/OWNER).

- FIA_AFL.1 because of the TSF detects Selectable (1-255) unsuccessful authentication attempts related to the verification of the PIN.

- FIA_UAU.1 because of the TSF allows export of configuration data except PIN and of D.IDData (User Data) as well as the function Digital Signature Sign on behalf of the user to be performed before the user is authenticated and also requires each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

- FIA_UAU.7 because of the TSF provides only information about invalid password length to the user while the authentication is in progress.

- FIA_UID.1 because of the TSF allows export of D.IDData (User Data) as well as the function Digital Signature Sign on behalf of the user to be performed before the user is identified and also requires each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

- FMT_MSA.1 because of the TSF enforces the change of the PIN to be restricted to the OWNER.

- FMT_MSA.2 because the TSF ensures that only secure values are accepted for security attributes.

- FMT_MSA.3 because of the TSF allows the OWNER to specify alternative initial values to override the default values when an object or information is created.

- FMT_SMF.1 because of the TSF is capable of performing the following security management functions: [Change PIN, Import User Data].

- FMT_SMR.1 because of the TSF maintains the roles CONFIGURATOR, USER and OWNER and is able to associate users with roles.

- FDP_ACC.2 because of the TSF enforces the Access Control SFP on D.KeyPriv, D.KeyPub, D.Signature, D.IDData and D.ConfData and all operations among subjects and objects covered by the SFP and therefore ensures that all operations between any subject in the TSC and any object within the TSC are covered by this access control SFP.

- FDP_ETC.1 because of the TSF enforces the D.KeyPub and D.IDData Export according to the Access Control SFP when exporting user data, controlled under the SFP(s), outside of the TSC which is done without the user data’s associated security attributes.

- FDP_ITC.1 because of the TSF enforces the D.IDData and D.ConfData Import according to the Access Control SFP when importing user data, controlled under the SFP, from outside of the TSC which is done by ignoring any security attributes associated with the user data when imported from outside the TSC and the import is only possible when token is in configuration mode during production. Afterwards it is only allowed after the token state has been elevated to an operational state of OWNER. OWNER state is reached through the valid execution of the VERIFY and CHANGE PIN command.

O.TOECryptoOP

The TOE shall provide cryptographic operations based upon a standard and state of the art algorithm. This will prevent an attacker from succeeding with a crypto analytical or mathematical attack. This is realized by:

- FCS_CKM.1 because of the TSF generates cryptographic keys in accordance with the specified cryptographic key generation algorithm [ECDSA] and specified cryptographic key sizes of 163 bit

- FCS_COP.1 because of the TSF performs ECDSA Sign with SHA-1 hashing in accordance with a specified cryptographic algorithm [ECDSA] and cryptographic key sizes of 163 bit

- FCS_RND.1 because the cryptographic operations for creation of the signature key pair and calculation of the signature are based on a quality metric for random numbers, which meet class K3 and SoF high (see [AIS20]).

- FPT_AMT.1 because the TSF runs a suite of tests during power-up (POST) to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

- FPT_TST.1 because of the TSF runs a suite of self tests during power-up to demonstrate the correct operation of the TSF and to verify the integrity of TSF data

O.TOEHardw

The TOE shall provide a secure hardware platform resistant against brute force hardware attacks. Furthermore the TOEs software functionality shall support the hardware measures in order to prevent direct hardware attacks. This is realized by:

- FCS_CKM.4 because of the TSF destroys cryptographic keys in accordance with a specified destruction method, which can be based on an assigned standard

- FIA_AFL.1 because of the TSF terminate the session establishment process after a specified number of unsuccessful user authentication attempts. It also requires that, after termination of the session establishment process, the TSF be able to disable the user account or the point of entry (e.g. workstation) from which the attempts were made until an administrator-defined condition occurs. By this, the token will be zeroised.

- FPT_AMT.1 because of the TSF runs a suite of tests during power-up to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF

- FPT_FLS.1 because of the TSF preserves a secure state when the following types of failures occur: over and under voltage, over and under clock frequency, temperature, electro magnetic radiation, physical probing, and firmware integrity problems.

- FPT_TST.1 because of the TSF runs a suite of self tests during power-up to demonstrate the correct operation of the TSF and to verify the integrity of TSF data

O.TOEKeyProtect

The TOE shall ensure that the private key is inaccessible other than for signing purposes. This is realized by

- FCS_CKM.1 because of the TSF generates cryptographic keys in accordance with the specified cryptographic key generation algorithm [ECDSA] and specified cryptographic key sizes of 163 bit

- FCS_RND.1 because the cryptographic operations for creation of the signature key pair and calculation of the signature are based on a quality metric for random numbers, which meet class K3 and the SoF high (see [AIS20]).

- FPT_AMT.1 because of the TSF runs a suite of tests during power-up to demonstrate the correct operation of the TSF.

- FPT_TST.1 because of the TSF runs a suite of self tests during power-up to demonstrate the correct operation of the TSF and to verify the integrity of TSF data

- FCS_COP.1 because of the TSF performs ECDSA Sign with SHA-1 hashing in accordance with a specified cryptographic algorithm [ECDSA] and cryptographic key sizes of 163 bit that means that the private key is accessible for signing purposes

- FDP_ACC.2 because the TSF enforces the Access Control SFP on D.KeyPriv which includes the inaccessibility for other purposes

- FDP_ETC.1 because of the TSF enforces the D.KeyPub and D.IDData Export according to the Access Control SFP when exporting user data, controlled under the SFP(s), outside of the TSC which is done without the user data's associated security attributes.

- FDP_ITC.1 because of the TSF enforces the D.IDData and D.ConfData Import according to the Access Control SFP when importing user data, controlled under the SFP, from outside of the TSC which is done by ignoring any security attributes associated with the user data when imported from outside the TSC and the import is only possible when token is in configuration mode during production. Afterwards it is only allowed after the token state has been elevated to an operational state of OWNER. OWNER state is reached through the valid execution of the VERIFY and CHANGE PIN command.

O.TOEPINblock

The TOE provides a PIN block functionality, which blocks the PIN Status after a reasonable secure amount of wrong attempts (e.g. after 3 attempts). This is realized by

- FDP_ACF.1 requirement because the Access Control SFP is based on the Token States,

- FIA_AFL.1 because of Selectable (1-255) unsuccessful authentication attempts related to the verification of the PIN are detected and the Token is zeroised afterwards

- FIA_UAU.1 because of the TSF allows export of configuration data except PIN and of D.IDData (User Data) as well as the function Digital Signature Sign on behalf of the user

to be performed before the user is authenticated and also requires each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

- FIA_UAU.7 because of the TSF provides only information about invalid password length to the user while the authentication is in progress.

- FIA_UID.1 because of the TSF allows export of D.IDData (User Data) as well as the function Digital Signature Sign on behalf of the user to be performed before the user is identified and also requires each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

- FMT_MSA.1 because of the change of the PIN is restricted to OWNER

- FMT_MSA.2 because the TSF ensures that only secure values are accepted for security attributes.

- FMT_MSA.3 because of the OWNER is possibly allowed to modify the PIN depended on the configuration.

- FMT_SMF.1 because of the TSF are capable of performing the security management functions *Change PIN* and *Import User Data*

- FMT_SMR.1 because of the TSF maintains the roles CONFIGURATOR, USER, OWNER and the TSF is able to associate users with roles.

8.2.2 Choice of TOE security assurance requirements

The choice of assurance requirements is based on the analysis of the security objectives for the TOE and on functional requirements defined to meet these objectives.

The assurance level is **EAL 4**.

Evaluation Assurance Level rationale

EAL 4 permits a developer to gain maximum assurance from security engineering based upon rigorous commercial development practices supported by moderate application of specialist security engineering techniques.

EAL 4 is therefore applicable for the DataSIGN(tm) product as users require a high level of independently assured security in a planned development and require a rigorous development approach without incurring unreasonable costs attributable to specialist security engineering techniques.

8.2.3 TOE Security requirements rationale

Functional requirements / Security Objectives	O.TOECryptoOP	O.TOEKeyProtect	O.TOEAuthent	O.TOEHardw	O.TOEPINblock
FCS_CKM.1	**X**	**X**			
FCS_CKM.4				**X**	
FCS_COP.1	**X**	**X**			
FCS_RND.1	**X**	**X**			
FDP_ACC.2		**X**	**X**		
FDP_ACF.1			**X**		**X**
FDP_ETC.1		**X**	**X**		
FDP_ITC.1		**X**	**X**		
FIA_AFL.1			**X**	**X**	**X**
FIA_UAU.1			**X**		**X**
FIA_UAU.7			**X**		**X**
FIA_UID.1			**X**		**X**
FMT_MSA.1			**X**		**X**
FMT_MSA.2			**X**		**X**
FMT_MSA.3			**X**		**X**
FMT_SMF.1			**X**		**X**
FMT_SMR.1			**X**		**X**
FPT_AMT.1	**X**	**X**		**X**	
FPT_FLS.1				**X**	
FPT_TST.1	**X**	**X**		**X**	

Table 21: 8.2.3 TOE Security requirements rationale

The Other attacker can be any person different from the User and the Configurator. The attack potential is considered to be low and the SOF as claimed in section 6.2 is low, too, and an appropriate claim for SOF.

8.2.4 IT-Environment security requirements rationale

Not applicable.

8.2.5 TOE Security functional requirement dependencies rationale

Family / Component / Element	Dependencies	Dependency fulfilled (yes/no)
FCS_CKM.1	FCS_COP.1 FCS_CKM.4 FMT_MSA.2	yes yes yes
FCS_CKM.4	FCS_CKM.1 FMT_MSA.2	yes yes
FCS_COP.1	FCS_CKM.1 FCS_CKM.4 FMT_MSA.2	yes yes yes
FCS_RND.1	None	-
FDP_ACC.2	FDP_ACF.1	yes
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	yes (ACC.2) yes
FDP_ETC.1	FDP_ACC.1	yes (ACC.2)
FDP_ITC.1	FDP_ACC.1 FMT_MSA.3	yes (ACC.2) yes
FIA_AFL.1	FIA_UAU.1	yes
FIA_UAU.1	FIA_UID.1	yes
FIA_UAU.7	FIA_UAU.1	yes
FIA_UID.1	None	-
FMT_MSA.1	FDP_ACC.1 FMT_SMF.1 FMT_SMR.1	yes (ACC.2) yes yes
FMT_MSA.2	ADV_SPM.1 FDP_ACC.1 FMT_MSA.1 FMT_MSR.1	yes yes (ACC.2) yes yes

Family / Component / Element	Dependencies	Dependency fulfilled (yes/no)
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	yes yes
FMT_SMF.1	None	-
FMT_SMR.1	FIA_UID.1	yes
FPT_AMT.1	None	-
FPT_FLS.1	ADV_SPM.1	yes
FPT_TST.1	FPT_AMT.1	yes

Table 22: 8.2.5 TOE Security functional requirement dependencies rationale

8.2.6 IT environment dependencies rationale

Not applicable.

8.2.7 TOE security assurance requirements and dependencies rationale

Requirement	Security Objective	Direct dependencies	Dependency fulfilled (yes/no)
ACM	**Configuration management (CM)**		
ACM_AUT.1	Partial CM automation	ACM_CAP.3	yes (ACM_CAP.4)
ACM_CAP.4	CM Capabilities	ACM_SCP.1 ALC_DVS.1	yes (ACM_SCP.2) yes
ACM_SCP.2	CM Scope	ACM_CAP.3	yes (ACM_CAP.4)
ADO	**Delivery and Operation**		
ADO_DEL.2	Delivery	ACM_CAP.3	yes (ACM_CAP.4)
ADO_IGS.1	Installation, generation and start-up procedures	AGD_ADM.1	Yes
ADV	**Development**		
ADV_FSP.2	Functional specification	ADV_RCR.1	yes (ADV_RCR.1)
ADV_HLD.2	Security enforcing high-level design	ADV_FSP.1 ADV_RCR.1	yes yes

Requirement	Security Objective	Direct dependencies	Dependency fulfilled (yes/no)
ADV_IMP.1	Implementation representation: Subset of the implementation of the TSF	ADV_LLD.1 ADV_RCR.1 ALC_TAT.1	yes yes (ADV_RCR.1) yes (ALC_TAT.1)
ADV_LLD.1	Descriptive low-level design	ADV_HLD.2 ADV_RCR.1	yes (ADV_HLD.2) yes (ADV_RCR.1)
ADV_RCR.1	Representation correspondence: Informal correspondence demonstration	None	-
ADV_SPM.1	Security policy model: Formal TOE security policy model	ADV_FSP.1	yes (ADV_FSP.2)
AGD	**Guidance documents**		
AGD_ADM.1	Administrator guidance	ADV_FSP.1	yes (ADV_FSP.2)
AGD_USR.1	User guidance	ADV_FSP.1	yes (ADV_FSP.2)
ALC	**Life cycle support**		
ALC_DVS.1	Development security: Identification of security measures	None	-
ALC_LCD.1	Life cycle definition: Developer defined life-cycle model	None	-
ALC_TAT.1	Tools and techniques: Well-defined development tools	ADV_IMP.1	yes (ADV_IMP.1)
ATE	**Tests**		
ATE_COV.2	Coverage: Analysis of coverage	ADV_FSP.1 ATE_FUN.1	yes (ADV_FSP.2) yes
ATE_DPT.1	Depth of testing Testing: low–level design	ADV_HLD.1 ATE_FUN.1	yes (ADV_HLD.2) yes yes
ATE_FUN.1	Functional testing	None	-

Requirement	Security Objective	Direct dependencies	Dependency fulfilled (yes/no)
ATE_IND.2	Independent testing	ADV_FSP.1	yes (ADV_FSP.2)
		AGD_ADM.1	yes
		AGD_USR.1	yes
		ATE_FUN.1	yes
AVA	**Vulnerability assessment**		
AVA_MSU.1	Analysis and testing for insecure states	ADO_IGS.1	yes
		ADV_FSP.1	yes (ADV_FSP.2)
		AGD_ADM.1	yes
		AGD_USR.1	yes
AVA_SOF.1	Strength of TOE security function evaluation	ADV_FSP.1	yes (ADV_FSP.2)
		ADV_HLD.1	yes (ADV_HLD.2)
AVA_VLA.2	Independent vulnerability analysis	ADV_FSP.1	yes (ADV_FSP.2)
		ADV_HLD.2	yes (ADV_HLD.2)
		ADV_IMP.1	yes (ADV_IMP.1)
		ADV_LLD.1	yes
		AGD_USR.1	yes

Table 23: TOE security assurance requirements and dependencies rationale

8.2.8 Mutually supportive and internally consistent rationale

The security functional requirements are complete and internally consistent because they are mutually supportive and provide an 'integrated effective whole' and all dependencies between security functional requirements are fulfilled. It is the same for security assurance requirements.

8.3 TOE summary specification rationale

8.3.1 Security functions rationale

The TOE Security Functions comprise the following functional requirements as of CC, Part 2 as well as FCS_RND (see 8.2.1.1).

TSF TOE Security Function	Security Functional Requirement
SF_KeyGenerate	FCS_CKM.1 FPT_AMT.1 (monitor interface to hardware protection)

TSF TOE Security Function	Security Functional Requirement
	FPT_TST.1 (PowerOnTest FIPS 140) FCS_RND.1
SF_UserAuthentication	FDP_ACF.1 FIA_AFL.1 FIA_UAU.1 FIA_UAU.7 FIA_UID.1 FMT_MSA.1 FMT_MSA.2 FMT_MSA.3 FMT_SMF.1 FMT_SMR.1
SF_SignatureCreation	FCS_COP.1 FCS_RND.1
SF_DataAccess	FDP_ACC.2 FDP_ETC.1 FDP_ITC.1
SF_AssetProtection	FCS_CKM.4 FIA_AFL.1 FPT_AMT.1 (monitor interface to hardware protection), FPT_FLS.1 FPT_TST.1 (PowerOnTest FIPS 140)

Table 24: Security functions rationale

SF_KeyGenerate provides public/private key generation based on the standard [ECDSA] and state of the art elliptic curves algorithm (ECDSA). The size of the generated keys will be 163 bit. The key pair generation will be initiated during production process and is not depending on the PIN verification status. In addition, SF_KeyGenerate can only be performed once during the power on self test (POST) in production phase.

Therefore SF_KeyGenerate has to fulfil the functional requirements:

FCS_CKM.1: The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [ECDSA] and specified cryptographic key sizes [163 bit] that meet the following: [ANSI X9.62 and FIPS 186 standards].

FCS_RND.1: The TSF shall provide a mechanism to generate random numbers that meet class K3 and the SoF high (see [AIS20]) in order to be used during the key pair generation.

FPT_AMT.1: The TSF shall run a suite of tests [during power-up (POST)] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

FPT_TST.1: The TSF shall run a suite of self tests [during power-up (POST)] to demonstrate the correct operation of the TSF.

All in all SF_KeyGenerate is suitable to meet the functionality of these SFR.

SF_UserAuthentication is accomplished by having the user entering a PIN, which is being validated against reference authentication data, which has been previously established.

Depending on the TOE configuration the user is allowed to change the PIN or not. If PIN change is possible the user needs to enter the correct PIN before PIN change is possible.

PIN block functionality is part of SF_UserAuthentication in order to ensure that the PIN related TOE functionality is being blocked after a reasonable number of wrong attempts (e.g. three attempts). The number of allowed attempts is to be set during configuration in production phase (the number of configurable possible attempts is limited to 255).

Therefore SF_UserAuthentication has to fulfil the functional requirements:

- FDP_ACF.1 because the TSF enforces the Access Control SFP based on Token States (CONFIGURATOR/USER/OWNER).

- FIA_AFL.1 because of the TSF detects Selectable (1-255) unsuccessful authentication attempts related to the verification of the PIN.

- FIA_UAU.1 because of the TSF allows export of configuration data except for the PIN and of D.IDData (User Data) as well as the function Digital Signature Sign on behalf of the user to be performed before the user is authenticated and also requires each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

- FIA_UAU.7 because of the TSF provides only information about Invalid Password Length to the user while the authentication is in progress.

- FIA_UID.1 because of the TSF allows export of D.IDData (User Data) as well as the function Digital Signature Sign on behalf of the user to be performed before the user is identified and also requires each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

- FMT_MSA.1 because of the TSF enforces the change of the PIN to be restricted to the OWNER.

- FMT_MSA.2. because of the TSF ensures that only secure values are accepted for security attributes.

- FMT_MSA.3 because of the TSF allows the OWNER to specify alternative initial values to override the default values when an object or information is created.

- FMT_SMF.1 because of the TSF is capable of performing the following security management functions: [Change PIN, Import User Data].

- FMT_SMR.1 because of the TSF maintains the roles CONFIGURATOR, USER and OWNER and is able to associate users with roles.

All in all SF_UserAuthentication is suitable to meet the functionality of these SFR.

SF_SignatureCreation provides a signature over supplied data, the current TOE PIN state and if configured also over the so called User Data (e.g. an account number). A standard and state of the art elliptic curves crypto algorithm (ECDSA) will be used by the TOE. The signature will be provided regardless of actual authentication state.

Therefore SF_SignatureCreation has to fulfil the following functional requirements:

FCS_COP.1 because of the TSF shall perform ECDSA Sign with SHA-1 hashing in accordance with a specified cryptographic algorithm [ECDSA] and cryptographic key sizes of 163 bit that meet the following: [ANSI X9.62, FIPS 186 standards].

FCS_RND.1: The TSF shall provide a mechanism to generate random numbers that meet class K3 and the SoF high (see [AIS20]) in order to be used during the signature creation.

All in all SF_SignatureCreation is suitable to meet the functionality of these SFR.

SF_DataAccess provides the ability to import/export User Data, Public Key and TOE Status depending upon the authenticated role of the user and the configuration settings done during production. Whereas the export will be provided regardless of actual authenticated role. The import of user data will only be available to successful authenticated roles.

Therefore SF_DataAccess has to fulfil the functional requirements:

- FDP_ACC.2 because of the TSF enforces the Access Control SFP on D.KeyPriv, D.KeyPub, D.Signature, D.IDData and D.ConfData and all operations among subjects and objects covered by the SFP and therefore ensures that all operations between any

subject in the TSC and any object within the TSC are covered by this access control SFP.

- FDP_ETC.1 because of the TSF enforces the D.KeyPub and D.IDData Export according to the Access Control SFP when exporting user data, controlled under the SFP(s), outside of the TSC which is done without the user data's associated security attributes.

- FDP_ITC.1 because of the TSF enforces the D.IDData and D.ConfData Import according to the Access Control SFP when importing user data, controlled under the SFP, from outside of the TSC which is done by ignoring any security attributes associated with the user data when imported from outside the TSC and the import is only possible when token is in configuration mode during production. Afterwards it is only allowed after the token state has been elevated to an operational state of OWNER. OWNER state is reached through the valid execution of the VERIFY and CHANGE PIN command.

All in all SF_DataAccess is suitable to meet the functionality of these SFR.

SF_AssetProtection zeroizes the private key, public key, user data and the PIN in case the TOE determines a physical attack (e.g. micro probing attacks).

Therefore SF_AssetProtection has to fulfil the functional requirements:

- FCS_CKM.4 because of the TSF destroys cryptographic keys in accordance with a specified destruction method which can be based on an assigned standard

- FIA_AFL.1 because of the TSF terminates the session establishment process after a specified number of unsuccessful user authentication attempts. It also requires that, after termination of the session establishment process, the TSF be able to disable the user account or the point of entry (e.g. workstation) from which the attempts were made until an administrator-defined condition occurs. By this, the token will be zeroised.

- FPT_AMT.1 because of the TSF runs a suite of tests during power-up to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF

- FPT_FLS.1 because of the TSF preserves a secure state when the following types of failures occur: over and under voltage, over and under clock frequency, temperature, electro magnetic radiation, physical probing, and firmware integrity problems.

- FPT_TST.1 because of the TSF runs a suite of self tests during power-up to demonstrate the correct operation of the TSF and to verify the integrity of TSF data

All in all SF_AssetProtection is suitable to meet the functionality of these SFR.

8.3.2 SOF rationale

SF_UserAuthentication

This SF uses a permutational mechanism for the authentication of the users (PIN code).

The strength of the function is <SOF-high>. The <SOF-high> for the authentication of the users is achieved with the following SFRs:

FIA_AFL.1 (The PIN is blocked after Selectable (1-255) consecutive failed authentication attempts).

SF_KeyGenerate and SF_SignatureCreation use a mechanism that provides random numbers. The security functions are claimed as class K3 and SoF high (see [AIS20]). This claim is achieved with the SFR FCS_RND.1 and the metric itself is described within the scope of [AIS20].

The others SFRs are not directly concerned by the PIN permutational mechanism.

8.3.3 Assurance measures rationale

Security assurance requirement	Assurance measure	Rationale
ACM		
ACM_AUT.1	AM_ACM	The assurance measure AM_ACM is about configuration management.
ACM_CAP.4	AM_ACM	The assurance measure AM_ACM is about configuration management, and confirms that the ACM_CAP.4 component is completed.
ACM_SCP.2	AM_ACM	The assurance measure AM_ACM is about configuration management, and confirms that the ACM_SCP.2 component is completed.
ADO		
ADO_DEL.2	AM_ADO	The assurance measure AM_ADO gives the delivery procedures and confirms that the ADO_DEL.2 component is completed.
ADO_IGS.1	AM_ADO	The assurance measure AM_ADO gives the installation, generation and start-up procedures and confirms that the ADO_IGS.1 component is completed.
ADV		
ADV_FSP.2	AM_ADV	The assurance measure AM_ADV gives the functional specification by describing the internal and external interfaces and confirms that the ADV_FSP.2 component is completed.

Security assurance requirement	Assurance measure	Rationale
ADV_HLD.2	AM_ADV	The assurance measure AM_ADV gives the architectural design by system decomposition and confirms that the ADV_HLD.2 component is completed.
ADV_IMP.1	AM_ADV	The assurance measure AM_ADV gives the implementation and confirms that the ADV_IMP.1 component is completed.
ADV_LLD.1	AM_ADV	AM_ADV The assurance measure AM_ADV gives the architectural design by subsystem decomposition and confirms that the ADV_LLD.1 component is completed.
ADV_RCR.1	AM_ADV	The assurance measure AM_ADV gives the correspondence demonstration and confirms that the ADV_RCR.1 component is completed.
ADV_SPM.1	AM_ADV	The assurance measure AM_ADV gives the security policy model and confirms that the ADV_SPM.1 component is completed
AGD		
AGD_ADM.1	AM_AGD	The assurance measure AM_AGD gives the administration documentation and confirms that the AGD_ADM.1 component is completed.
AGD_USR.1	AM_AGD	The assurance measure AM_AGD gives the user documentation and confirms that the AGD_USR.1 component is completed.
ALC		
ALC_DVS.1	AM_ALC	The assurance measure AM_ALC gives the security measures and confirms that the ALC_DVS.1 component is completed.
ALC_LCD.1	AM_ALC	The assurance measure AM_ALC gives the development process and confirms that the ALC_LCD.1 component is completed.
ALC_TAT.1	AM_ALC	The assurance measure AM_ALC gives the development tools and confirms that the ALC_TAT.1 component is completed.
ATE		
ATE_COV.2	AM_ATE	The assurance measure AM_ATE gives the test documentation and confirms that the ATE_COV.2 component is completed.

Security assurance requirement	Assurance measure	Rationale
ATE_DPT.1	AM_ATE	The assurance measure AM_ATE gives the test documentation and confirms that the ATE_DPT.1 component is completed.
ATE_FUN.1	AM_ATE	The assurance measure AM_ATE gives the test documentation and confirms that the ATE_FUN.1 component is completed.
ATE_IND.2	AM_ATE	The assurance measure AM_ATE gives the test documentation and confirms that the ATE_IND.2 component is completed.
AVA		
AVA_MSU.2	AM_AVA	The assurance measure AM_AVA gives the misuse analysis and confirms that the AVA_MSU.2 component is completed.
AVA_SOF.1	AM_AVA	The assurance measure AM_AVA gives the SOF evaluation and confirms that the AVA_SOF.1 component is completed.
AVA_VLA.2	AM_AVA	The assurance measure AM_VLA gives the vulnerability analysis and confirms that the AVA_VLA.2 component is completed.

Table 25: Assurance measures rationale

8.3.4 Mutually supportive and internally consistent rationale

The IT security functions are complete and internally consistent because they are mutually supportive and provide an 'integrated effective whole'. The interactions between security functions are limited to the dependencies between these security functions. It is the same for assurance measures.

8.4 PP claims rationale

Not applicable.

9 Annex

9.1 Standard (CC) Abbreviations & Glossary

ADM	Administrator guidance
AIS	Application notes and Interpretation of the Scheme
AUT	CM Automation
BSI	Bundesamt für Sicherheit in der Informationstechnik (Federal office for Security in IT)
CAP	CM Capabilities
CC	Common Criteria (referenced as CC)
COV	Coverage of testing
DEL	Delivery
DPT	Depth of testing
DVS	Development Security
EAL	Evaluation Assurance Level
FSP	Functional Specification
FUN	Functional tests
HLD	High Level Design
IGS	Installation, Generation and Start-up
IMP	Implementation
IND	Independent testing
LCD	Life Cycle Definition
LLD	Low Level Design
MSU	Misuse
PP	Protection Profile
RCR	Representation Correspondence
SCP	CM Scope
SFP	Security Function Policy
SFR	Security Functional Requirement
SOF	Strength of security Functions
SPM	Security Policy Modelling
ST	Security Target
TAT	Tools And Techniques
TOE	Target of Evaluation
TSF	TOE Security Function
USR	User guidance

VLA	Vulnerability Analysis

9.2 Specific Abbreviations & Glossary

APDU	Application Protocol Data Units,
CRC	Cyclic Redundancy Check
EC	Elliptic Curve
ECDSA	Elliptic Curve Digital Signature Algorithm
FDC	First Data Corp.
PIN	Personal Identification Number
POST	Power On Self Test
Private key	Key generated by the TOE during POST, used for generation of the signature
Public key	Key generated by the TOE during POST, exported to outside the TOE, used for verification of the signature outside the TOE
RNG	Random Number Generator
SEF	Security Function provided by the HW (SLE66C42P)
SHA-1	Secure Hash Algorithm
XML	Extensible Markup Language

9.3 References

[AIS 20] Application Notes and Interpretation of the Scheme (AIS) AIS 20, Version 1, Date: 2 December 1999, Status: Mandatory, Subject: Functionality classes and evaluation methodology for, deterministic random number generators, Publisher: Certification body of the BSI, Section II 2, as part of the certification scheme

[CC] Common Criteria for Information Technology Security Evaluation, version 2.1, Annotated with interpretations as of 2003-12-31, August 1999
Part 1: Introduction and general model, CCIMB-99-031,
Part 2: Security functional requirements, CCIMB-99-032,
Part 3: Security Assurance Requirements, CCIMB-99-032.

[ECDSA] Working Draft AMERICAN NATIONAL STANDARD X9.62-1998 Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA) September 20, 1998 Accredited by the American National Standards Institute

[FIPS 140-2] SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES May 25, 2001 Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8900

[FIPS 186-2] FIPS PUB 186-2 (+Change Notice), FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, 2000 January 27, U.S. DEPARTMENT OF

COMMERCE/National Institute of Standards and Technology, DIGITAL SIGNATURE STANDARD (DSS)

[SC_2030] Certicom Source Code Files, 2.0.3.0, Certicom

[ST_IC] Infineon Technologies AG, Security and Chipcard ICs, Evaluation Documentation, SLE66C82P/SLE66C42P / m1474/m1495 / A14, Security Target, Version 1.3, Date 21-03-2005, Authors: Walter, Novinsky, Buchmüller

[X9.62] American National Standard, X9.62-1998, Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), November 1998.