INSIKA
TIM Interface Specification
V.2.1.0

Revision: 00

Last change: 27.06.2017

Status: released

This document is a translation of the original German version. In case of any discrepancy between the original text and this translation, the German text shall prevail.

About this document

This INSIKA documentation contains the specifications of the working group INSIKA of the ADM e.V. for the TIM interface. The publication of the documentation enables interested parties to find information on the principles of the INSIKA technology, and are thus in a position to implement the system in practice. To be able to inform about modifications, the documentation will only be distributed to registered companies.

No guarantees or obligations can be derived from the technical contents presented in this description. The authors and the consortium assume no liability or responsibility if the system or parts thereof are adopted and/or implemented.

In-development support for implementations is currently not available. Questions about the system will be answered as far as possible.

Further information: http://www.insika.de/

The INSIKA project was funded by the German Federal Ministry of Economics and Technology under the grant MNPQ 11/07.

Revision History

Version	Date	Change
0.9	30.05.2008	
0.9.27	10.03.2009	Publication by distribution to all registered companies
T106-02	09.03.2010	Publication to registered companies
T110-01	06.08.2014	Review and update, integration of supplement for Version 1.1.0
T110-02	19.03.2015	Cash register profile moved to separate document, various details clarified, Review
V.2.0.0-00	14.12.2016	Publication to registered companies (German version only) Changes compared to the previous version: • Alternative signature length ECDSA-256 with SHA-256, New transaction command for special applications, • New command GET LATEST RESPONSE, • Signature of Null turnovers possible • Changes to the handling of turnovers from third party and delivery note
V.2.1.0-00	27.06.2017	Publication to registered companies Changes compared to the previous version: • Integration of additional error codes • Indication of long Reports (>256 Byte) • Extension of GET DATA TIM Status extended • New command GET DATA Memory Status

Authors:	Mathias Neuhaus (cryptovision GmbH) Jörg Wolff (PTB [until 2013]) Norbert Zisky (ADM, PTB [until 2015]))
Title:	INSIKA TIM Interface Specification
Referenced TIM package:	V.2.1.0
Revision:	00
Status:	Released
Last change:	27.06.2017
File name:	INSIKA_TIM_Interface-V210-en.docx
Number of pages:	88
Contact:	http://www.insika.de/en/contact

Table of Contents

List of tables

List of illustrations

List of abbreviations

Abbre-viation	Explanation
AID	Application IDentifier (ISO 7816-5)
APDU	Application Protocol Data Unit (ISO 7816)
ASCII	American Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation One
ATR	Answer to Reset (ISO 7816-3)
BCD	Binary Coded Decimal
BER	Basic Encoding Rules (ASN.1)
BP	here: Booking position (= ITEM)
CLA	Class byte, part of the ISO 7816 command APDU
DER	Distinguished Encoding Rules (ASN.1)
DF	Dedicated File (ISO 7816-4)
ECDSA	Elliptic Curve Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
EF	Elementary File (ISO 7816-4)
FID	File Identifier (ISO 7816-4)
ID	IDentification
INS	Instruction, part of the ISO 7816 command APDU
ITEM	Item, booking position (= BP)
LC	Length command, part of the ISO 7816 command APDU
LE	Length expected, part of the ISO 7816 command APDU
MF	Master File (ISO 7816-4)
MSB	Most Significant Bit
OID	Object Identifier

Abbre-viation	Explanation
P1, P2	Parameters 1 and 2, part of the ISO 7816 command APDU
PIN	Personal Identification Number
PIX	Proprietary application Identifier eXtension (ISO 7816-5)
RID	Registered IDentifier (ISO 7816-5)
SFID	Short File ID
SHA-1	Secure Hash Algorithm (FIPS 180-1)
SHA-256	Secure Hash Algorithm (FIPS 180-4)
SW1, SW2	Status words 1 and 2 (ISO 7816)
TIM	Tax Identification Module
TLV	Tag Length Value (BER-TLV)
TP ID	Tax Payer IDentification
VAT ID	Value-added tax ID number
VAT	Value Added Tax
WID	Business Identification Number

Glossary

ANSI X9.62	American National Standards Institute: Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA)
Authorized body	Organisation that has been assigned with the task of card issuance. This includes the role of a Certificate Authority (CA = CA), i.e. processing applications, creating and managing the cryptographic certificates etc. as well as special services like providing access to the tax authorities to the database
Booking	A sales transaction usually consisting of several transaction items, is executed with the command TRANSACTION
Certificate	here: Electronic certificate, file signed by an authorised body, see also X.509v3
Container	here: Composite TLV object for turnover data
Daily closing (report)	is executed with the command REPORT
Delivery note	Sale of goods or the provision of services for which an invoice is issued later but that should nevertheless be documented in a secure form
FIPS 180	National Institute of Standards and Technology (NIST): Federal Information Processing Standards Publication 180, Secure Hash Standard (SHS), (i.a. definition of SHA-1)
FIPS 180-4	National Institute of Standards and Technology (NIST): Federal Information Processing Standards Publication 180, Secure Hash Standard (SHS), (see definition for SHA-256)
FIPS 186	National Institute of Standards and Technology (NIST): Federal Information Processing Standards Publication 186, Digital Signature Standard (DSS), (i.a. definition of ECDSA)
Hash, hash value	Result of a cryptographic one-way function, or more precisely: Cryptographic hash function, in ECDSA-192 mode SHA-1 and in ECDSA-256 mode SHA-256
INSIKA	Project name, INSIKA = INtegrierte SIcherheit für messwertverarbeitende Kassensysteme = Integrated security solution for cash register systems processing measured values
ISO/IEC 7816	International Organization for Standardization and the International Electrotechnical Commission, Information Technology – Identification cards – Integrated circuit(s) cards with contacts
ITU-T X.509v3	International Telecommunication Union - Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks (standard for electronic certificates)

ITU-T X.690	International Telecommunication Union - Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), (also standardised in ISO/IEC 8825-1)
Negative turnover	Negative portions of the turnover of a booking. Negative turnovers are transmitted as positive numbers.
Personalisation	Conducted by the issuing authorised body before the issurance of the TIM, encompasses setting the data in accordance with → 8.1, generation of the key pair on the TIM, issurancee and application of the certificate
Positive turnover	Positive portions of the turnover of a booking. Positive turnovers are not transmitted separately. The following relationship is true: Turnover = positive turnover - | negative turnover |
Private key	Private (and confidential) part of the key pair of the TIM, is used for signature generation, non-legible, never leaves the TIM
Profile	here: Definition geared to a special field of application that includes the mapping of accounting items to data structures (→ 1.3)
Public key	Public part of the key pair of the TIM required for signature verification
REPORT	Command to the TIM with which the totalising memories of the TIM are sent back. The command performs a "daily closing".
Sequence number	Strictly monotonously ascending number assigned by the TIM
Signature	Electronic signature, is calculated by the TIM
Smart card	Chip card without the software necessary for INSIKA (TIM package)
Tax Identification Module (TIM)	Smart card with INSIKA software ("TIM package")
Third-party turnover	Third-party turnovers are sales of goods or services on behalf of and for the account of a third party, which are printed on the receipt nonetheless. During a tax audit it must be proved that these sales are subject to tax for the third-party.
TIM package	Software package on the smart card that provides the special INSIKA functions
TLV object	here: Data object where the actual data ("value") are preceded by one byte for unambiguous identification ("tag") and one byte as a length field ("length") (see also "Use of the Basic Encoding Rules" in ISO/IEC 7816-4 Annex D).
Totalising memories	Memories for various amounts and quantities on the TIM which store totals. Reports (command REPORT) are based on the totalising memories.

Training	Transactions for training or test purposes. They are not taxable sales.
TRANSACTION	Command to the TIM with which the data of a booking are transmitted to the TIM. After a positive plausibility check, a signature is returned.
Turnover	VAT-related remuneration for a transaction (and hence the sum of the individual prices of the items of a transaction); is always related to a VAT class; can be transmitted to the TIM as gross or net turnover; monetary unit
Turnover overflow	If the maximum capacity of a totalising memory should be exceeded (impossible in normal operation) the memory contains a value which is too low (example: assuming a maximum value of 1000 the attempt to store 1015 would result in 15). To make the situation recognisable the overflow will be indicated when reading the value or any total based on this value.
Value Added Tax (VAT)	here: (Value-added) tax share of the remuneration for a transaction; here monetary unit
VAT class	Definition independent of current VAT rates, represents the currently valid value-added tax rate
VAT rate	Percentage rate of the value-added tax

1 General Information

1.1 Changes

1.1.1 Changes to version T.1.1.0

This version of the specification was developed based on the following requirements:

- Opening up other areas of application for INSIKA
- Adaptation to current cryptographic algorithms and key lengths
- Improved mapping of special circumstances (delivery notes and third party turnover)

Changes to the data model were not made for compatibility reasons and were also not required. In comparison to the previous version (T.1.1.0), the significant changes described here were made.

Due to specific requirements for the desired areas of application, additional variants of the TRANSACTION command were defined. The VERIFY command was also appropriately enhanced.

There is no longer a focus on a single signature algorithm. In addition to the previous parameters (ECC, 192 bit NIST, SHA-1), a further set of parameters is available (ECC, 256 bit NIST, SHA-256). Selection is made at the time of personalisation. The variant used in each specific case can be (indirectly) determined via the command GET DATA Hash Length or directly via GET DATA Cryptographic Algorithms.

A significant change affects the handling of third party turnover. These are now transmitted based on the VAT rate together in the turnover containers E1h…E6h; transmission in the previous container E7h no longer takes place. The container E7h is only managed by the TIM using the transmitted turnovers. However, there are no changes at all for all those application cases that have not used turnover from third party turnover up to now! The INSIKA profiles for cash registers and taximeters have been adapted where required.

In addition, errors that have been identified have been corrected and some sections have been formulated more precisely.

1.1.2 Changes to version 2.0.0

The maximum possible length on a report command requires support of „extended length“ APDUs, see ISO 7816-4. Therefore the report 98 D1h was introduced. Farther the error codes 98 D2h and 98 F2h were added. If an error 98 E1h, 98 E2h or 98 F2h occurs the TIM will be deactivated automaticly. The commands GET DATA TIM Status extended and GET DATA Memory Status were specified. All changes and additions have no effects to existing implementations.

1.2 Document Information

This document describes the interface of the INSIKA smart card, hereinafter referred to as "TIM" (Tax Identification Module). This document provides the knowledge required to integrate the TIM into a cash register system and other application environments. For this purpose, a profile specification adapted to the specific application environment is available in a separate document.

The change to the designation for the TIM interface documentation from “T” to “V” is due to the switchover from the test phase to real operation. The TIM cards T.xx have been successfully used since 2010 in a variety of application areas. No problems with either hardware or software occured.

An understanding of smart card interfaces and the terminology used in this field is assumed. The document presents only the information necessary for implementation of the TIM interface. The internal operation of the TIM is not specified in this document. Knowledge of the INSIKA system is necessary in order to understand this documentation.

1.2.1 Structure of the Document

The document is structured as follows:

Chapter 1 contains general information on the document and on the smart card.

Chapter 2 documents the data objects transmitted via the interface, first in general terms and then in the order of their structure (tag / length / value).

Chapter 3 presents the INSIKA commands that can be transmitted to the TIM.

Chapter 0 contains the error codes that can be given back in the response to a command from the TIM.

Chapter 5 explains the life cycle of the TIM.

Chapter 6 contains definitions of value-added tax classes, character substitution and rounding.

Chapter 7 presents all the information necessary for signature verification.

Chapter 8 explains all the data stored on the TIM.

Chapter 9 explains the system of the INSIKA profiles.

The Annex (→ 10) contains examples and corresponding explanations.

Chapters 1 - 6 and the document about the corresponding profile are the basic preconditions for a basic integration of the INSIKA technology into a system. If a complete implementation is to be carried out, chapters 7 and 8 also have to be observed.

1.2.2 Language

This document is a translation of the original German version. The German version already uses English for designators of data objects (tags), commands and error codes – they remain unchanged.

1.2.3 Hexadecimal Representation

The hexadecimal representation of bytes is indicated by the appending of a lowercase "h": xxh, xx xxh are hexadecimal representations of the byte xx or the bytes xx xx.

1.2.4 Use of Prefixes

The prefix "TIM" designates data objects used in direct communication with the TIM, i.e. these data objects are used directly in the field "Data" of the ISO 7816 APDUs.

Data objects with other prefixes (e.g. "ITEM") are defined in profiles and are not used directly on the TIM interface. The only exception here is the optional command HASH with which any data object can be transmitted.

1.3 Cryptographic algorithms

For data security purposes, the INSIKA system applies digital signatures to the data to be protected using the the ECDSA algorithm as specified in ANSI X9.62. SHA-1 in accordance with FIPS 180-1 with 160 bits and ECDSA with a key length of 192 bits are currently used as the hash functions. The INSIKA cards (TIM) supplied in accordance with specification version T.1.1.0 only support these key lengths.

INSIKA cards supplied in accordance with this specification support larger key lengths with ECDSA-256 and SHA-256. The switchover to longer key lengths was based on the recommendation of the German Federal Office for Information Security (BSI). The TIM cards supplied in accordance with this specification support both key lengths. However, it is necessary to state the key length when ordering the cards for technical reasons. Following the production and delivery of the cards, the signature key length cannot be changed during their lifespan. There is thus a differentiation between ECDSA-192 and ECDSA-256 mode.

1.4 Concept of the INSIKA Profiles

INSIKA profiles serve to map application-specific data of a system to data structures. A profile defines the data objects over which the hash value of the transaction items is calculated. During a transaction, this hash value (→ 2.6.9) is transmitted to the TIM and signed (→ 3.4).

The data objects of a profile (transaction items) are not transmitted directly to the TIM during a transaction. As the hash value of these data objects is signed, however, these data objects are also (indirectly) included into the signature. A large number of data objects can thus be included into the signature without these having to be transmitted to the TIM interface. The TIM remains unchanged in all cases when the profile is changed.

Through the concept of the profiles it is possible to adapt the INSIKA system to various applications. In particular, a TIM can be used to protect the data of a wide variety of systems. Exactly one profile is used in each application.

Chapter 9 contains further information about the concept of profiles. An implementation always requires the separate documentation for the profile used in the application.

1.5 Format of the Smart Card

The TIM is supplied in the format ID-1 according to ISO 7816. The smart card is perforated in such a way that it can turned into the ID-000 or 3FF (MicroSim) format by breaking out. The location of the contacts is defined in ISO 7816-2.

1.6 Electrical Properties

The electrical properties of the TIM follow ISO 7816-3. The TIM V.2.1.0 is a smart card of Class A and B, and can therefore be used in the range Vcc = 2.7 to 5.5 V (see ISO 7816-3).

1.7 Transmission Protocol / Card Reader

The data are transmitted to and from the TIM according to the T=1 protocol (ISO 7816-3). The support of „extended length“ APDUs is nessescary. The TIM can thus be operated using PC card readers, ISO 7816 interface ICs or directly. For PC card readers, a Class 1 model is sufficient, a PIN pad is not necessary.

A physical integration of the card reader (with the TIM) into the respective system is not required. Thus, for example, communication with the TIM via a (wireless) network is possible to use INSIKA on platforms that do not allow the integration of a card reader.

2 Data Objects

2.1 TLV Encoding

All data that are stored in field "Data" of the ISO 7816 APDUs are transmitted in TLV format. For this the actual data (VALUE → 2.6) are preceded by one byte for unambiguous identification (TAG → 2.4) and one byte for the length (LENGTH → 2.5). The whole data object is referred to hereinafter as TLV object. The BER-TLV used here is identical to the "SIMPLE-TLV".

2.2 Composite Data Objects

Data objects can be composed of several sub-elements. Composite data objects are referred to as "constructed DO" to distinguish them from simple data objects ("primitive DO"). Composite data objects can be described in short as "nested TLV objects".

Figure 2-1 shows a composite data object containing three data objects. LC indicates here the overall length of all the simple data objects contained (here: P1, P2 and P3) including the corresponding TL encoding.

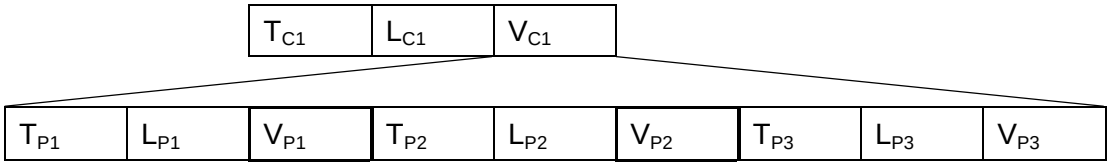

Figure 2-1: Composite data object

Composite data objects are used with the TIM for the transmission of turnovers. "Containers" form the "constructed DO" here and the turnover data contained the "primitive DOs"

2.3 Data Types

The data types according to Table 2-1 are used for the “payload data”, i.e. in the field VALUE of the TLV objects. The number of octets used depends on the particular TLV object and is defined in section → 2.6.

Table 2-1: Definition of the data types

Designation of the data type	Value range (for an octet)		Remarks
	hexadecimal	Decimal	
binary	00h..FFh	0..255	Unsigned
ASCII	20h..7Fh	32..127	Printable characters (further limited in some cases by the character substitution → 6.2)
unsigned BCD	00h..99h	0..99	Here only "Packed BCD" is used
Signed BCD	9Dh..0Ch..9Ch	-9..0..+9	Applies only to the least significant byte, all other bytes by analogy with type "unsigned BCD"

2.3.1 Binary

Binary values are transmitted unsigned with the most significant bit first (MSB first).

If the length of the payload data is defined as variable, leading zero bytes have to be suppressed. If the length of the payload data is defined as fixed, leading zero bytes must not be suppressed.

2.3.2 ASCII

Text contents are transmitted in ASCII code. Only the value range of printable characters from 20h to 7Fh is used here. In order to be able to recover printed characters, this value range is further limited in certain data objects by a character substitution (→ 6.2).

2.3.3 Unsigned BCD

BCD values of type "unsigned BCD" are transmitted as packed BCD (i.e. with two decimal digits per byte). The byte of the most significant digit is transmitted first. With an odd number of places, the most significant nibble of the first byte is zero (e.g. 0xh yzh).

2.3.4 Signed BCD

BCD values of type "signed BCD" are transmitted as packed BCD (i.e. with two decimal digits per byte). The byte of the most significant digit is transmitted first. With an odd number of digits, the most significant nibble of the first byte is zero (e.g. 0xh yzh).

The sign is stored in the least significant nibble. The sign is encoded as C – positive or D – negative; all other encodings (0..B, E, F) are invalid. The value zero is defined with a positive sign: 0Ch.

Leading zero bytes are not permitted and must be suppressed.

2.4 TAG (designator)

The tags for the TLV objects of the TIM are defined in Table 2-2 (listed in ascending order according to the numerical value of the tag).

Table 2-2: Summary of the defined TLV tags

Name	Tag	Description	Payload data		Transm.-direction	
			Length (Byte)	Data type (→ 2.3)	Request	Response
TIM_SIGNATURE	9Eh	TIM signature (transaction or report) → 2.6.1	48/ 64[1]	binary	X[2]	X

[1] In terms of the cryptographic algorithm, it can be selected when ordering the INSIKA cards whether the cards should work a) with ECDSA-192/SHA-1 (signature 48 bytes, hash value: 20 bytes) or b) with ECDSA-256/SHA-256 (signature: 64 bytes, hash value: 32 bytes). Variant b) is the standard configuration for the new generation of cards.

[2] Is transmitted to the TIM only when using the optional command VERIFY SIGNATURE (→ 3.6).

Name	Tag	Description	Payload data		Transm.-direction	
			Length (Byte)	Data type (→ 2.3)	Request	Response
TIM_LIFECYCLE	C0h	TIM life cycle → 2.6.2	1	binary		X
TIM_SERIAL_NO	C1h	TIM serial number → 2.6.3	*[3]	*		X
TIM_VERSION	C2h	Version of the TIM application, → 2.6.4	7.. 10	ASCII		X
TIM_TRANSPORT_PIN [4]	C3h	TIM transport PIN → 2.6.5	6	ASCII	X	
TIM_TP_ID	C4h	TIM Tax Payer IDentification → 2.6.6	1.. 32	ASCII	X	X
TIM_TP_ID_NO	C5h	Consecutive number of the TIM referred to a TP_ID (Tax Payer IDentification Number) → 2.6.7	1..4	binary	X	X
TIM_OPERATOR	C6h	TIM operator ID → 2.6.8	1.. 16	ASCII	X	
TIM_HASH_ TRANSACTION_ITEMS	C7h	TIM hash value of the transaction items → 2.6.9	20/ 32[5]	binary	X	
TIM_CURRENCY	C8h	TIM currency code → 2.6.11	2	binary	X	
TIM_VAT_ADDON	C9h	TIM flag “VAT not included” → 2.6.13	0	-	X	
TIM_TRAINING	CAh	TIM flag “training” → 2.6.13	0	-	X	
TIM_SEQ_NO_ TRANSACTION	CBh	TIM sequence number of a transaction → 2.6.14	1..4	binary	X[6]	X
TIM_SEQ_NO_ REPORT	CCh	TIM sequence number of a signed report → 2.6.14	1..4	binary		X

[3] Length and format of the serial number depends on the underlying smart card.

[4] Use of the transport PIN is optional. Depending on whether or not the transport PIN is used, the TIM is delivered to the tax payer in life cycle TIM_PERSONALISED or TIM_ACTIVATED.

[5] The length of the hash value is dependent on the mode, see footnote 1

[6] Is transmitted to the TIM only when using the optional command VERIFY SIGNATURE (→ 3.6).

Name	Tag	Description	Payload data		Transm.-direction	
			Length (Byte)	Data type (→ 2.3)	Request	Response
TIM_DATE	CDh	TIM date → 2.6.15	3 / 4	un-signed BCD	X	X
TIM_TIME	CEh	TIM time → 2.6.16	2	un-signed BCD	X	X
TIM_MONTH	CFh	List of months, coded as offset to the first bookable month → 2.6.17	0..12 1	binary		X
TIM_MONTH_START	D0h	TIM date, first month of a turnover period → 2.6.15	3	un-signed BCD	X	
TIM_MONTH_END	D1h	TIM date, last month of a turnover period → 2.6.15	3	un-signed BCD	X	
TIM_SEQ_NO_ TRANSACTION_FIRST	D2h	TIM sequence number of the first transaction of a turnover period → 2.6.14	1..4	binary		X
TIM_SEQ_NO_ TRANSACTION_LAST	D3h	TIM sequence number of the last transaction of a turnover period → → 2.6.14	1..4	binary		X
TIM_HASH_ REPORT_ITEMS	D4h	TIM hash value of the report items → 2.6.10	20/ 32[7]	binary	X	
TIM_COUNTRY_CODE	D5h	Country code → 2.6.12	2	binary		X
TIM_ TURNOVER _ THIRDPARTY	D6h	Third party turnover → 2.6.18	1..6	signed BCD	X	
TIM_ DELIVERYNOTE	D7h	TIM flag delivery note → 2.6.13	0	-	X	
TIM_TURNOVER	D8h	TIM turnover → 2.6.18	1..6 ..11[8]	signed BCD	X	X
TIM_TURNOVER_ NEGATIVE	D9h	TIM negative turnover → 2.6.18	1..6 ..11[9]	signed BCD	X	X

[7] The length of the hash value is dependent on the mode, see footnote 1

[8] In the request 1..6 byte length, in the response 1..11 byte length

[9] In the request 1..6 byte length, in the response 1..11 byte length

Name	Tag	Description	Payload data		Transm.-direction	
			Length (Byte)	Data type (→ 2.3)	Request	Response
TIM_TURNOVER_ VAT	DAh	TIM value-added tax → 2.6.18	1..6	signed BCD	X	X
TIM_TURNOVER_ VAT_RATE	DBh	TIM value-added tax rate → 2.6.19	1..2	un-signed BCD	X	X
TIM_TURNOVER_ COUNTER	DCh	TIM transaction counter → 2.6.20	1..4	binary		X
TIM_TURNOVER_ OVERFLOW	DDh	TIM flag “turnover overflow” → 2.6.12	0	-		X
TIM_TURNOVER_ VAT_CHANGED	DEh	TIM flag “change of VAT rate” → 2.6.13	0	-		X
TIM_CONTAINER_ VAT_1 … TIM_CONTAINER_ VAT_6	E1h .. E6h	TIM container 1..6 (corresponding to VAT rate 1..6) → 2.6.21	6.. 34	(TLV-Obj.)[10]	X	X
TIM_CONTAINER_ THIRDPARTY	E7h	TIM container third-party transaction → 2.6.21	3.. 23	(TLV obj.)	X	X
TIM_CONTAINER_ DELIVERYNOTE	E8h	TIM container delivery note → 2.6.21	3.. 23	(TLV obj.)	X	X
TIM_CONTAINER_ TRAINING	E9h	TIM Container Training → 2.6.21	6.. 21	(TLV obj.)		X

The data fields required or permitted per command and their sequence is specified in the individual explanations in chapter → 2.6.

2.5 LENGTH

The following encoding is used for the length of the TLV objects:

The length of the payload data field is directly encoded in one byte (00h..7Fh).

In the current version of the TIM specification, no fields with lengths longer than 127 (7Fh) are used. The BER-TLV used here is identical with the "SIMPLE-TLV".

2.6 VALUE

This chapter defines the data contents to be used for the payload data of the TLV objects. The formats used internally on the TIM for storage may deviate from these and are docu-

[10] Containers are composite data objects. They can contain TLV objects D8h, D9h, DAh, DBh, DCh and DEh. See → 2.2.

mented by the manufacturer of the TIM application. These internal formats are not visible at the TIM interface.

The TLV objects with the prefix "TIM" are used in commands and responses on the TIM interface.

2.6.1 TIM Signature: TIM_SIGNATURE – 9Eh

The signature is transmitted as a sequence of two 192/256-bit binary numbers (MSB first) (i.e. in total 384/512 bits = 48/64 bytes), see footnote 1. The two numbers are transmitted in direct succession in a single TLV object. Leading zero bytes are not suppressed.

The signature is generated on the TIM for transactions or reports according to the respective hash specification (→ 7.1 and → 7.2).

Example for ECDSA-192

T	L	V	Contents
9Eh	30h	4Ah 07h 1Ch 49h 3Fh 4Fh BDh 3Eh 8Dh 0Dh FBh 1Ah D4h 98h D6h DBh DEh 71h BEh E2h 92h 5Eh 51h FAh B1h 31h 89h B9h 92h 4Eh 96h 1Fh 2Dh 7Ch 18h 3Ch 20h 83h B0h 65h 33h 99h EDh 6Dh D2h 8Ch F0h D6h	Signature

Example for ECDSA-256

T	L	V	Contents
9Eh	40h	4Ah 07h 1Ch 49h 3Fh 4Fh BDh 3Eh 8Dh 0Dh FBh 1Ah D4h 98h D6h DBh DEh 71h BEh E2h 92h 5Eh 51h FAh B1h 31h 89h B9h 92h 4Eh 96h 1Fh 2Dh 7Ch 18h 3Ch 20h 83h B0h 65h 33h 99h EDh 6Dh D2h 8Ch F0h D6h 01h 43h 3Eh 22h 1Ch 5Dh 11h 71h 89h D6h 3Ch A4h 7Fh 22h 4Dh 1Bh	Signature

2.6.2 TIM Life Cycle: TIM_LIFECYCLE – C0h

The life cycle indicates the current status of the TIM and is stored on the TIM. The life cycle is returned as a TLV object with the payload data length of one byte. The encoding of the life cycle and the life cycle transitions are described under → 5.

Example

T	L	V	Contents
C0h	01h	03h	TIM is activated

2.6.3 TIM serial number: TIM_SERIAL_NO – C1h

The serial number of the TIM is an unambiguous identification number assigned by the manufacturer of the underlying smart card. The format of this serial number is defined by the smart card manufacturer and cannot be specified here.

2.6.4 TIM version: TIM_VERSION – C2h

The version number of the TIM application is encoded in the format

A.V.R.M

with one letter and three sequences of numbers (each max. two digits) separated by dots. The length of the TLV object TIM Version is 7..10 bytes. The elements of the version number have the meanings described in Table 2-3.

Table 2-3: Definition of the TIM version number

	Meaning	Description
A	Application	'T' for test version, 'V' for version in the live phase
V	Version	Version of the TIM specification
R	Release	Compatible extensions of the specification
M	Maintenance	Error corrections only

In terms of the relevantly used cryptographic algorithm, a TIM corresponds to exactly one version & release (V.R) of the specification.

Example

T	L	V	Contents
C2h	07h	54h 2Eh 31h 2Eh 31h 2Eh 30h	"T.1.1.0"
C2h	07h	56h 2Eh 32h 2Eh 31h 2Eh 30h	"V.2.1.0"

2.6.5 TIM Transport PIN: TIM_TRANSPORT_PIN – C3h

The transport PIN consists of exactly 6 characters that are set during personalisation of the TIM. The transport PIN is encoded as ASCII numbers (30h..39h). For activation of the TIM (→ 3.5.4), the transport PIN is transmitted in direct succession without suppression of leading zeros.

Example

T	L	V	Contents
C3h	06h	30h 31h 32h 33h 34h 35h	'0' '1' '2' '3' '4' '5'

2.6.6 TIM tax payer ID: TIM_TP_ID – C4h

The tax payer ID serves for tax identification of the data signed with the cash register and the TIM. The business identification number (WID), if assigned, can be used for this. Alternatively, the value-added tax identification number (VAT ID) can be used. The tax payer ID together with the consecutive number of the TIM (→ 2.6.7) forms an unambiguous identifying characteristic. It is possible that the tax authorities will impose more concrete stipulations.

During the personalisation of the TIM, the tax payer ID is set on the TIM. It is defined with 1..32 ASCII characters in the value range 21h..7Eh. Space characters are not permitted. After a TRANSACTION (all relevant variants) or REPORT command, the tax payer ID - C4h and the sequence number of the TIM - C5h is appended to the data set to be signed and returned in the respective response.

Example

T	L	V	Contents
C4h	0Ch	44h 45h 30h 31h 32h 33h 34h 35h 36h 37h 38h 39h	Tax payer ID: "DE0123456789"

2.6.7 TIM sequence number: TIM_TP_ID_NO – C5h

The consecutive number of the TIM (per tax payer ID) is stored with 1-4 bytes in unsigned binary format on the TIM and transmitted. Leading zero bytes are omitted.

Example

T	L	V	Contents
C5h	02h	01h A0h	Sequence number of the TIM: 416

2.6.8 TIM operator ID: TIM_OPERATOR – C6h

The identification of the operator is transmitted with 1..16 ASCII characters in the value range 21h..7Eh. The transmitted data are validated as to their value range by the TIM application. If the payload data field contains characters outside the value range 21h..7Eh, the error message 98 04h TIM_ERROR_INVALID_CHARACTER is returned.

The operator ID must be prepared by the cash register according to the character substitution before transmission to the TIM (→ 6.2). The value range is thereby further limited.

Example

T	L	V	Contents
C6h	10h	6Ch 69h 73h 65h 6Ch 6Fh 74h 74h 65h 23h 62h 65h 72h 73h 63h 68h	'l' 'i' 's' 'e' 'l' 'o' 't' 't' 'e' '#' 'b' 'e' 'r' 's' 'c' 'h' after character substitution (Original: "Liselotte Überschwang" becomes liselotte#berschwang and is then shortened to 16 characters)

2.6.9 TIM hash value of the transaction items: TIM_HASH_TRANSACTION_ITEMS – C7h

The hash value of the transaction items is transmitted as a 20/32 byte binary value (MSB first), see footnote 1. Leading zero bytes are not suppressed.

The hash value of the transaction items is the result of the profile used (→ 9). Depending on the profile, the hash value of the transaction items is calculated according to the respective hash specification of the profile.

Example for SHA-1 (20 Byte)

T	L	V	Contents
C7h	14h	5Bh 3Ch 07h E5h A9h 4Eh 34h 9Bh 2Dh BDh EEh C4h 4Bh 3Eh B9h FDh 62h 98h F0h 33h	Hash value of the transaction items

Example for SHA-256 (32 Byte)

T	L	V	Contents
C7h	20h	D3h 2Bh 56h 8Ch D1h B9h 6Dh 45h 9Eh 72h 91h EBh F4h B2h 5Dh 00h 7Fh 27h 5Ch 9Fh 13h 14h 9Bh EEh B7h 82h FAh C0h 71h 66h 13h F8h	Hash value of the transaction items

2.6.10 TIM hash value of the report items: TIM_HASH_REPORT_ITEMS – D4h

The hash value of the report items is transmitted as a 20/32 byte binary value (MSB first), see footnote 1. Leading zero bytes are not suppressed.

The hash value of the report items is the result of the profile used. Depending on the profile, the hash value of the report items is used and calculated according to the respective hash specification of the profile.

Example for SHA-1 (20 Byte)

T	L	V	Contents
D4h	14h	5Bh 3Ch 07h E5h A9h 4Eh 34h 9Bh 2Dh BDh EEh C4h 4Bh 3Eh B9h FDh 62h 98h F0h 33h	Hash value of the report items

Example for SHA-256 (32 Byte)

T	L	V	Contents
D4h	20h	D3h 2Bh 56h 8Ch D1h B9h 6Dh 45h 9Eh 72h 91h EBh F4h B2h 5Dh 00h 7Fh 27h 5Ch 9Fh 13h 14h 9Bh EEh B7h 82h FAh C0h 71h 66h 13h F8h	Hash value of the report items

2.6.11 TIM currency code: TIM_CURRENCY – C8h

The currency code is set on the TIM during personalisation. The currency code is transmitted to the TIM at each transaction. The TIM compares the two codes and in the event of a negative result, sends back the error 98 13h TIM_ERROR_CURRENCY (→ 4).

The currency code is defined according to ISO 4217, whereby the numerical code is used here. The code is transmitted as a 2 byte binary value (MSB first) – leading zeros are **not** suppressed. The Euro has the currency code 978 or 03 D2h.

Example

T	L	V	Contents
C8h	02h	03h D2h	Euro currency code: 978

2.6.12 TIM country code: TIM_COUNTRY_CODE – D5h

The country code is entered on the TIM during personalisation. It serves to differentiate between different countries with the same currency code.

Remark: The country code is optional. The personalisation office decides whether to enter the country code.

The country code is defined according to ISO 3166, whereby the numerical code is used here. The code is transmitted as a 2 byte binary value (MSB first) – leading zero bytes are **not** suppressed.

For example, the country code for Germany is 276 or 01 14h.

Example

T	L	V	Contents
D5h	02h	01h 14h	Country code for Germany: 276

2.6.13 TIM flag excl. tax, training, turnover overflow, change of VAT rate: TIM_VAT_ADDON – C9h, TIM_TRAINING – CAh, TIM_DELIVERYNOTE – D7h, TIM_TURNOVER_OVERFLOW – DDh, TIM_TURNOVER_VAT_CHANGED – DEh

Flags (for example overflow) are encoded as TL objects without VALUE (payload data). The length field LENGTH contains the length 00h.

A set flag is indicated purely by the presence of the TL object. Flags that are not set will not be transmitted.

Example

T	L	V	Contents
C9h	00h	-	Flag “VAT not included” set
CAh	00h	-	Flag “Training” set
D7h	00h		Flag “Deliverynote” set
DDh	00h	-	Flag “Turnover overflow” set
DEh	00h	-	Flag “Change of VAT rate” set

2.6.14 TIM sequence numbers: TIM_SEQ_NO_TRANSACTION – CBh, TIM_SEQ_NO_REPORT – CCh TIM_SEQ_NO_TRANSACTION_FIRST – D2h TIM_SEQ_NO_TRANSACTION_LAST – D3h

The sequence numbers for transactions and reports are transmitted from the TIM as unsigned 32 bit binary values (MSB first). The length of the TLV object is variable from 1 to 4 bytes. Leading zero bytes are suppressed.

Two sequence number counters are defined on the TIM: One for signed transactions (→ 3.4) and one for signed reports (→ 3.5.1). The sequence number counters are incremented only when a signature is created.

The two sequence numbers TIM_SEQ_NO_TRANSACTION_FIRST and TIM_SEQ_NO_TRANSACTION_LAST are returned in the response from a report. These indicate the sequence numbers of the first and last transaction in the turnover period.

Example

T	L	V	Contents
CBh	02h	04h D2h	Transaction sequence number: 1234
CCh	01h	0Fh	Report sequence number: 15
D2h	03h	01h E2h 40h	Sequence number of the first transaction of the turnover period: 123456
D3h	04h	07h 5Bh CDh 15h	Sequence number of the last transaction of the turnover period: 123456789

2.6.15 TIM date:

TIM_DATE – CDh
TIM_MONTH_START – D0h
TIM_MONTH_END – D1h

Dates are transmitted in the formats

YYYYMMDD or

YYYYMM

as unsigned BCD. YYYY indicates the four-digit year, MM the two-digit month and DD the two-digit day.

The format used depends on the length of the payload data (4 or 3 bytes). The date TIM_DATE with a length of 3 bytes is used with the GET DATA commands (→ 3.2.2 and → 3.2.3).

The TLV objects TIM_MONTH_START and TIM_MONTH_END define the beginning and end of a turnover period in the format YYYYMM. The length is also 3 bytes.

All other commands use the date with the length of 4 bytes.

Date inputs are checked by the TIM application for the correct format and value range (max. period of validity of the TIM).

Example

T	L	V	Contents
D0h	03h	20h 09h 02h	'2' '0' '0' '9', '0' '2'
CDh	04h	20h 09h 02h 24h	'2' '0' '0' '9', '0' '2', '2' '4'

2.6.16 TIM time:

TIM_TIME – CEh

Times are transmitted in the format

hhmm

in 24-hour format as unsigned BCD.

hh is the two-digit hour and mm the minutes. The length of the TLV object is 2 bytes. The value ranges from 00 00h to 23 59h.

Example

T	L	V	Contents
CEh	02h	23h 09h	'2' '3' '0' '9'

2.6.17 TIM month: TIM_MONTH – CFh

List of months, every month is transmitted as the offset to the first month to which transactions can be booked. A is transmitted as an unsigned 8 bit binary value. The list can be empty. In that case the tag (CFh) and the length zero (00h) are transmitted only.

The concrete meaning of a month list is described at the commands (see also → 3.2.3 and 3.2.6).

Example

T	L	V	Contents
CDh	03h	20h 08h 01h	Date "2008","01" → 01/2008 is the first month to which transactions can be booked
CFh	04h	00h 0Ah 0Dh 0Eh	Month 0, 10, 13, 14 Offset to the first month of validity → List contains the months 01/2008, 11/2008, 02/2009 and 03/2009

The tag CDh indicates the first month of validity for the TIM.

2.6.18 TIM turnover, negative turnover, value-added tax and third party turnover: TIM_TURNOVER – D8h, TIM_TURNOVER_NEGATIVE – D9h, TIM_TURNOVER_VAT – DAh TIM_TURNOVER_THIRDPARTY –D6h

Turnovers and value-added tax are transmitted in the data type signed BCD (see → 2.3.4). Leading zero bytes must be suppressed.

All values are scaled to the smallest currency unit (here: Euro Cents) so that no decimal points are used.

Turnovers and value-added taxes with the value null are not allowed to be transmitted. In these cases, the relevant TLV object has to be omitted.

The TLV object Turnover indicates the resulting total turnover of the transaction per VAT class. Third part turnovers must be included in the total turnover and, if relevant, the total negative turnover for the VAT class. Negative turnovers reduce the turnover and the corresponding VAT amount. Negative turnovers are shown in the TLV object Negative turnover and transmitted as a positive number.

The turnover for a transaction may also be negative. If the turnover is negative, a negative turnover must be transmitted to the TIM that is larger than or equal to the absolute amount of

the turnover. Larger negative turnovers occur if the turnover results from positive and negative transaction items.

Turnovers and negative turnovers identify the VAT-related total amounts for a transaction. These must not be confused with the prices of individual transaction items defined in profiles

The third party turnover must contain all the third party transaction items for a VAT rate. This total is added together internally by the TIM as gross turnover in the container third part turnover. Turnover, negative turnover, value-added tax and third party turnover are always transmitted for a transaction request in one container (→ 2.6.21).

Examples

Examples of valid and invalid encodings (currency unit: Euro cents):

T	**L**	**V**	**Contents**
D8h	02h	12h 3Ch	Turnover: 1.23 €
D8h	03h	01h 23h 4Dh	Turnover: -12.34 €
DAh	01h	1Ch	Value-added tax 0.01 €
D9h	04h	01h 00h 00h 0Ch	Negative turnover: 1000.00 €
D8h	06h	00h 00h 00h 00h 12h 3Ch	**Invalid**, leading zero bytes not suppressed
D8h	03h	34h 56h 70h	Invalid sign
D9h	01h	0Dh	**Invalid** sign for zero
D6h	02h	12 3Dh	Third party turnover: -1.23 €

2.6.19 TIM value-added tax rate: TIM_TURNOVER_VAT_RATE – DBh

VAT rates are transmitted in the data type unsigned BCD with 2 or 4 valid places (see → 2.3.3).

All values are standardised to the smallest unit 0.01%. No decimal point are therefore used.

A leading zero byte must be suppressed. A VAT rate of 0% must be transmitted.

Examples

Examples of valid and invalid encodings:

T	L	V	Contents
DBh	02h	19h 00h	VAT rate: 19%
DBh	02h	05h 50h	VAT rate: 5.5%
DBh	02h	07h 00h	VAT rate: 7%
DBh	01h	00h	0%, VAT-free
DBh	02h	00h 10h	**Invalid**, leading zero byte not suppressed

2.6.20 TIM transaction counter: TIM_TURNOVER_COUNTER – DCh

The respective counters of the transactions for third-party, delivery note and training memory are managed on the TIM and returned as unsigned 32 bit binary values (MSB first) (see → 2.3.3).

Leading zero bytes are suppressed. The TLV object has a variable length of 1..4 bytes.

Examples

T	L	V	Contents
DCh	02h	0Bh 00h	Transaction counter: 2816
DCh	03h	01h FFh 01h	Transaction counter: 130817

2.6.21 TIM container 1…6 and container for third-party, delivery note and training: TIM_CONTAINER_VAT_1…6 – E1h…E6h, TIM_CONTAINER_THIRDPARTY – E7h, TIM_CONTAINER_DELIVERYNOTE – E8h, TIM_CONTAINER_TRAINING – E9h

The containers 1..6 and the containers third-party (→Glossary, 8.2.2), delivery note (→Glossary, 8.2.3) and training (→Glossary, 8.2.4) with tags E1h..E9h are composite data objects (→ 2.2).

They can contain the following TLV objects:

- Turnover, negative turnover, value-added tax (tags D8h.. DAh → 2.6.18),
- Third party turnover (tag D6h) for a transaction request
- VAT rate (tag DBh → 2.6.19)
- transaction counter (tag DCh → 2.6.20),
- flags for turnover overflow and change of VAT rate (tags DDh, DEh → 2.6.12).

The tags for container 1..6, container for third-party, delivery note and training indicate the memories on the TIM in which the turnovers are to be updated (→ 8.2). Depending on the VAT class, the corresponding container 1..6 has to be used. The assignment is defined in section → 6.1.1.

Only the data objects D6h, D8h, D9h, DAh and DBh are permitted for the transmission direction to the TIM. In the direction from the TIM, responses of the REPORT can also contain the data objects DCh..DEh (→ 2.6.12, 2.6.20). The containers E7h and E8h contain the gross totals managed by the TIM for transactions from third party turnover and delivery notes.

If, in a container, turnover and negative turnover have the value zero, this container is not transmitted.

The validity of the transmitted values is checked by the TIM. This check encompasses the correct encoding and, with negative turnovers, the comparison with the turnover – if the turnover is negative, the negative turnover must be larger than or equal to the absolute amount of the turnover.

Example: Turnover = -100 € → Negative turnover must be larger than / equal to 100 €

Examples

Container 1 with standard VAT rate, turnover = +11.99 €, (negative turnover = 0 € is eliminated), VAT = +1.91 €, VAT rate = 19.00%

T_C	L_C	V_C												
		T_{P1}	L_{P1}	V_{P1}			T_{P2}	L_{P2}	V_{P2}		T_{P3}	L_{P3}	V_{P3}	
E1h	0Dh	D8h	03h	01h	19h	9Ch	DAh	02h	19h	1Ch	DBh	02h	19h	00h

Product return -11.99 €, purchase +11.99 €, both 7% VAT
Container 2 with reduced VAT rate 7%, (turnover = +11.99 € - 11.99 € = 0 € eliminated), negative turnover = +11.99 €, (VAT = -0.78 € + 0.78 € = 0 € eliminated), VAT rate = 7,00 %

T_C	L_C	V_C								
		T_{P1}	L_{P1}	V_{P1}			T_{P2}	L_{P2}	V_{P2}	
E2h	09h	D9h	03h	01h	19h	9Ch	DBh	02h	07h	00h

3 Commands

The commands described in Table 3-1 are available at the TIM interface:

Table 3-1: Commands of the TIM application

Command	Encoding CLA INS P1 P2	Options
SELECT FILE	00h A4h 0xh 0yh	x, y encoding in accordance with ISO 7816-4 → 3.1
GET DATA	00h CAh 01h yyh	yy = F0 – TIM status yy = F1 – TIM status, extended yy = F2 – Booked Months yy = F3 – Hash length yy = F4 – Cryptograhic algorithms yy = F5 – Memory Status → 3.2
READ CERTIFICATE	00h B0h xxh yyh	ISO 7816-4 command READ BINARY xx, yy encoding in accordance with ISO Short file IDs are not supported → 3.3
TRANSACTION	80h 40h xxh 00h	xx = 00 TRANSACTION xx = 01 - TR Data xx = 02 - TR Tax Payer xx = 03 - TR Time Stamp → 3.4
REPORT	80h 42h xxh 00h	xx = 01 – Signed xx = 02 – Unsigned xx = 03 – Span xx = 04 – TIM Activate xx = 05 – TIM Deactivate → 3.5
GET LATEST RESPONSE	80h C0h 00h 00h	→ 3.6
VERIFY SIGNATURE	80h 44h 00h 00h	→ 3.7
HASH	00h 2Ah 90h 80h or 10h 2Ah 90h 80h	ISO 7816-8 command PSO_H → 3.8

The commands GET DATA and REPORT offer options that are appended to the name, e.g. "GET DATA TIM Status". The commands are explained in more detail below. A result code is returned with each command (→ 4).

Note: The encodings in this chapter provide useful values for LE - not the ones valid for ISO 7816 only

3.1 SELECT FILE

The SELECT FILE command selects the TIM application or one of the associated files. It is the standard ISO 7816-4 command. The possible options for P1 and P2 are defined in ISO 7816-4.

3.1.1 Selection of the TIM Application

The TIM application is installed as the “default selected” application on the smart card. Therefore, all commands for the TIM application are immediately available after switching on the supply voltage – or after a RESET.
It is not necessary to explicitly select the application – but it is possible. This ensures compatibility with previous cards.

The TIM application can be selected via the registered Application ID (AID). The AID for ECDSA-192 mode consists of the Registered Identifier (RID)[11]: **D2h 76h 00h 01h 48h** plus the Proprietary application Identifier eXtension (PIX): **'T' 'I' 'M' (**54h 49h 4Dh**)**. For ECDSA-256 mode, the RID: D2h 76h 00h 01h 72h plus PIX **'T' 'I' 'M’** is used. The variant of the TIM application is defined during the personalisation of the TIM. Only the personalised variant (196 bit **or** 256 bit) can be selected in each case.

The length of the transmitted data is indicated in the LC field. P2 = 0Ch suppresses the return of values so that the length of the expected response (LE) does not have to be specified, but it is possible too. The complete code for the command is therefore:

Command

CLA	INS	P1	P2
00h	A4h	04h	0Ch

Data field for ECDSA-192 mode

LC	Data	LE
08h	D2h 76h 00h 01h 48h 54h 49h 4Dh	--

Data

AID (D2h 76h 00h 01h 48h 54h 49h 4Dh)

Data field for ECDSA-256 mode

LC	Data	LE
08h	D2h 76h 00h 01h 72h 54h 49h 4Dh	--

Data

AID (D2h 76h 00h 01h 72h 54h 49h 4Dh)

Response

SW1 / SW2	
67 00h	LC invalid
6A 82h	File / application not found
6A 86h	P1 / P2 invalid
90 00h	No error

[11] http://www.kartenbezogene-identifier.de/de/rapi/rid-liste.html

3.1.2 Selection of a File

A file is selected via the file ID (FID). The length of the transmitted data is indicated in the LC field. P2 = 0Ch suppresses the return of values so that the length of the expected response (LE) does not have to be specified.

The following example shows the selection of the certificate file. The certificate is stored on the TIM in the file **EF_CERT** with the file ID **11h 10h**. The complete code for the command for selection of the file EF_CERT is therefore:

Command

CLA	INS	P1	P2
00h	A4h	00h	0Ch

LC	Data	LE
02h	11h 10h	--

Data

File ID (here: EF_CERT = 11h 10h)

Response

SW1 / SW2	
67 00h	LC invalid
6A 82h	File / application not found
6A 86h	P1 / P2 invalid
90 00h	No error

3.2 GET DATA

The GET DATA command reads information via the TIM application. This command expands the ISO 7816 command with the same name.

The data field of this command is empty. Only the length of the expected response has to be encoded in the LE field (00h → all data).

The command offers three variants that are distinguished by the encoding of P2. The variants are explained in detail in the following sections.

Command

CLA	INS	P1	P2
00h	CAh	01h	see Table 3-2

LC	Data	LE
--	--	00h

Table 3-2: GET DATA Parameter P2

P2	Meaning
F0h	GET DATA TIM Status → 3.2.1
F1h	GET DATA TIM Status, extended → 3.2.2
F2h	GET DATA TIM Booked Months → 3.2.3
F3h	GET DATA Hash Length → 3.2.4
F4h	GET DATA Cryptographic Algorithms→3.2.5
F5h	GET DATA Memory Status → 3.2.6

Response

See following sections → 3.2.1 - 0

3.2.1 GET DATA TIM Status

In response to the command GET DATA with P2 = F0h, the TIM transmits brief status information.

Command

CLA	INS	P1	P2
00h	CAh	01h	F0h

LC	Data	LE
--	--	00h

Response

The following data objects are returned TLV encoded:

Tag	Length (Byte)	Value
C0h	1	Life cycle of the TIM → 2.6.2

SW1 / SW2	
67 00h	LC invalid
6A 86h	P1 / P2 invalid
90 00h	No error

3.2.2 GET DATA TIM Status extended

In response to the command GET DATA with P2 = F1h, the TIM transmits extended status information.

Command

CLA	INS	P1	P2
00h	CAh	01h	F1h

LC	Data	LE
--	--	00h

Response

The data objects listed in Table 3-3 are returned TLV encoded.

Table 3-3: TLV Response with Extended TIM Status

Tag	Length (Byte)	Value
C0h	1	Life cycle of the TIM → 2.6.2
C2h	7..10	TIM version → 2.6.4
C4h	1..32	Tax payer ID → 2.6.6
C5h	1..4	Consecutive number of the TIM → 2.6.7
C1h	*	Serial number of the TIM → 2.6.3
C8h	2	Currency code → 2.6.10
D5h	2	Country code → 2.6.12, if configured
CDh	3	Date (of the last possible transaction month) → 2.6.15
CBh	1..4	Sequence number for transactions (current) → 2.6.14
CCh	1..4	Sequence number for reports (current) → 2.6.14
CDh	4	Date of personalization (LC ≥ 02) → 2.6.15
CDh	4	Date of activation (LC ≥ 03) → 2.6.15
CDh	4	Date of deactivation (LC ≥ 04) → 2.6.15
02h	1	Kind of deactivation (LC ≥ 04) → Table 3-4

* The length of the serial number depends on the smart card used and is specified by the manufacturer.

SW1 / SW2	
67 00h	LC invalid
6A 86h	P1 / P2 invalid
90 00h	No error

The fiedls „ date of personalization“, date of deactivation“ and „kind of deactivation“ are only returned in the respectively life cycle of the card.

The field „kind of deactivation“ indicates whereby the TIM has been deactivated, see siehe Table 3-4:

Table 3-4: Kind of deactivation

Code	Kind of deactivation
00h	Deactivation with REPORT TIM Deactivate
01h	Memory error
02h	Bad entry of PIN
03h	Internal error
04h … 7Fh	Reserved for TIM specification.
80h … FFh	Usable by TIM supplier.

3.2.3 GET DATA TIM Booked Months

In response to the command GET DATA with P2 = F2h, the TIM transmits a list of the months to which turnover has been booked to date.

Command

CLA	INS	P1	P2
00h	CAh	01h	F2h

LC	Data	LE
--	--	00h

Response

The data objects listed in Table 3-5 are returned TLV encoded.

Table 3-5: TLV Response with List of Turnover Months

TAG	Length (Byte)	Value
CDh	3	First month to which turnover can be booked → 2.6.15
CFh	0..121	List of the months with turnover, a month with turnover is transmitted as a binary offset to the first month to which turnover can be booked (→ CDh) → 2.6.17

The months to which turnover has been booked to date are transmitted as an offset to the month to which the first transaction can be booked. The number of months already booked is therefore identical to the length of the payload data field.

An example of this response can be found under → 2.6.17.

SW1 / SW2	
67 00h	LC invalid
6A 86h	P1 / P2 invalid
90 00h	No error

3.2.4 GET DATA Hash Length

In response to the command GET DATA with P2 = F3h, the TIM transmits the length of the supported hash values for TRANSACTION and REPORT commands.

Command

CLA	INS	P1	P2
00h	CAh	01h	F3h

LC	Data	LE
--	--	00h

Response

The data objects listed in Table 3-6 are returned TLV encoded:

Table 3-6: TLV Response with Length of the Hash Value

TAG	Length (Byte)	Value
02h	1	Length of the hash value in bytes

SW1 / SW2	
67 00h	LC invalid
6A 86h	P1 / P2 invalid
90 00h	No error

3.2.5 GET DATA Cryptographic Algorithms

In response to the command GET DATA with P2 = F4h, the TIM transmits the OIDs for the cryptographic algorithms used.

Command

CLA	INS	P1	P2
00h	CAh	01h	F4h

LC	Data	LE
--	--	00h

Response

The data objects listed in Table 3-7– in the stated sequence – are returned.

Table 3-7: TLV Response with Length of the OID

TAG	Length (Byte)	Value
06h	l_{OID}	OID of the signature algorithm (ECDSA with SHA-1 / ECDSA with SHA-256)
06h	l_{OID}	OID of the underlying curve (192 / 256 bit)
06h	l_{OID}	OID of the hash algorithm for the transaction item (SHA-1 / SHA-256)

The OIDs, see Table 3-8, are all transmitted with the same tag 06h (according to the ASN.1 tag for OIDs); the meaning is derived from the position and the value. The length of the signature can be determined from the second response tag based on the OID of the curve. The length of hash value for the transaction item to be transmitted is determined from the OID for the hash algorithm of the third tag. Table 3-8 shows the OID in ASN.1 dot notation in line 1 and the OID in ASN.1 DER encoding in line 2 in each case.

SW1 / SW2	
67 00h	LC invalid
6A 86h	P1 / P2 invalid
90 00h	No error

Table 3-8: Object identifiers used by TIM

OID	Meaning
1.2.840.10045.4.1 06 07 2A8648CE3D0401	Signature algorithm: ECDSA with SHA-1
1.2.840.10045.4.3.2 06 08 2A8648CE3D040302	Signature algorithm: ECDSA with SHA-256
1.2.840.10045.3.1.1 06 08 2A8648CE3D030101	ECC curve: NIST 192bit (random)
1.2.840.10045.3.1.7 06 08 2A8648CE3D030107	ECC curve: NIST 256bit (random)
1.3.14.3.2.26 06 05 2B0E03021A	Hash algorithm: SHA-1
2.16.840.1.101.3.4.2.1 06 09 608648016503040201	Hash algorithm: SHA-256

3.2.6 GET DATA Memory Status

The TIM transmits the state of the internal memory with the command GET DATA mit P2 = F5h. That is usable for the concrete analysis of a memory error. An evaluation of the stored data via the command REPORT is still possible.

Command

CLA	INS	P1	P2
00h	CAh	01h	F5h

LC	Data	LE
--	--	00h

Answer

The listed data objects in Table 3-9 are returned TLV coded.

The date of month (CDh) could be wrong in case of an error of the configuration memory. The list of months with errors of memory is still correct.

Table 3-9: TLV Answer with status of memory

TAG	Length (Byte)	Value
01h	1	00h: **no** error in configuration memory FFh: **error** in configuration memory
CDh	3	Moment of the first month when booking is possible → 2.6.15
CFh	0..121	List of month with memory error. A month with memory error could be transmitted as binary offset to the first month (→ CDh) where booking is possible → 2.6.17

SW1 / SW2	
67 00h	LC invalid
6A 86h	P1 / P2 invalid
90 00h	no error

3.3 READ CERTIFICATE

The command READ CERTIFICATE reads the certificate stored on the TIM. The certificate contains details of the tax payer and the public key of the TIM and is stored in format X.509v3. The information in the certificate on the TIM are a subset of Table 8-1.

The command READ CERTIFICATE corresponds to the ISO 7816-4 command READ BINARY. Thus the encoding of the parameters P1 and P2 corresponds to the ISO standard.

The certificate is stored on the TIM in the elementary file EF_CERT with the file ID 11h 10h. The file to be read must be selected via the SELECT FILE command (selection of the certificate, see → 3.1.2). Short file IDs (SFID) are not supported.

ISO 7816-3 defines responses with max. 254 bytes information length. Depending on the configuration, the TIM can return longer responses. Due to the limitation, the certificate has to be read in several steps. In the first step, a part of the data is read. In further steps, the rest of the data can be read by specifying the offset.

The data field of the command is empty. The length of the data to be read must be encoded in the LE field (00h → maximum response length). In order to simplify the calculation, for example, a length of 128 bytes (80h) can be used here.

An example of reading out the certificate can be found under → 10.1.4.

Command

CLA	INS	P1	P2
00h	B0h	Xx	Yy

LC	Data	LE
--	--	*

* Length of the requested data

Parameter P1 / P2

Encoding P1 (xx) High byte (Offset)
Encoding P2 (yy) Low byte (Offset)

The maximum offset can be 32767 (decimal) / 7FFFh (hex).

Response

Data
Requested data

SW1 / SW2	
xx xxh	See error messages (→ 4) and ISO 7816-4
90 00h	No error

3.4 TRANSACTION

The command TRANSACTION transmits the data of a transaction or a freely selectable data block to the TIM.

When signing any data freely selected by the host, the data set to be signed in the INSIKA environment must always be linked at least to the sequence number generated internally by the card. Only in this way is it possible to verify the completeness of the recorded data. In order to verify the signed data set, all of the data elements added by the TIM application for the relevant transaction command are required.

The length of the transmitted data is indicated in the LC field (marked with *). The length of the expected response has to be encoded in the LE field (00h → all data).

Command

CLA	INS	P1	P2
80h	40h	00h, 01h / 02h / 03h	00h

LC	Data	LE
*	Data (see → 3.4.1 – 3.4.2)	00h

* Length of the transmitted data

Parameter P1

P1	Meaning
00h	TRANSACTION
01h	TR Data
02h	TR Tax Payer
03h	TR Time Stamp

3.4.1 TRANSACTION (booking)

The TIM carries out a plausibility check on the received data for transactions[12]. In the case of a positive result, the TIM generates an unambiguous sequence number, updates the internal totalising memory, prepares the data according to the TRANSACTION hash specification (→ 7.1.1), signs the data set and returns the signature. In the case of a negative result, an error code (→ 4) is returned. Before a transaction, the hash value of the transaction items has to be calculated. The specifications for this can be found in the definition of the corresponding profile (→ 1.4and → 9).

Command

CLA	INS	P1	P2
80h	40h	00h	00h

LC	Data	LE
*	Transaction data → Table 3-10	00h

* Length of the transmitted transaction data

Data

The transaction data contain the TLV objects according to Table 3-10.

[12] The TIM carries out a semantic, syntactic and content-based plausibility check for the transmitted data structure and turnover data.

Table 3-10: Transaction request

Tag	Length (Byte)	Value	possibly not transmitted[13]	signature-relevant[14]
CDh	4	Date → 2.6.15		X
CEh	2	Time → 2.6.16		X
C6h	1..16	Operator ID → 2.6.8		X
C7h	20/32	Hash value of the transaction items → 2.6.9		X
C8h	2	Currency code → 2.6.11	X[15]	
C9h	0	Flag "VAT not included" → 2.6.13	X	X
CAh	0	Flag "Training mode" → 2.6.13	X	X
D7h	0	Flag "Delivery note" → 2.6.13	X	X
E1h	6..28	Container 1 → 2.6.21	X	
D8h	1..6	Turnover → 2.6.18		X
D9h	1..6	Negative turnover → 2.6.18	X	
DAh	1..6	Value-added tax → 2.6.18	X	
DBh	1..2	VAT rate → 2.6.19		X
D6h	1..6	Third party turnover → 2.6.18	X	X
		…	X	
E6h	6..28	Container 6 → 2.6.21	X	
D8h	1..6	Turnover → 2.6.18		X
D9h	1..6	Negative turnover → 2.6.18	X	
DAh	1..6	Value-added tax → 2.6.18	X	
DBh	1..2	VAT rate → 2.6.19		X
D6h	1..6	Third party turnover Container third-party → 2.6.18	X	X

Remark: All turnovers or values that are not null must be transmitted; in the case of null turnovers or flag not set, the data object is omitted.
Training turnovers are marked with the flag "Training mode" (tag CAh → 2.6.12).

[13] Null turnovers and flags not set are not transmitted.

[14] Data object goes directly into the signature

[15] In the case of transaction requests with turnovers not equal to null, C8h - currency code needs to be transmitted.

If the flag "VAT not included" (→ 2.6.12) was not transmitted, the turnovers and negative turnovers are treated as gross values. If the flag "VAT not included" is transmitted, the turnovers and negative turnovers are treated as net values.

Response after Positive Plausibility Check

In the event of a positive plausibility check[16] of the transaction, the tax payer ID as well as the consecutive number of the TIM (saved on the TIM during personalisation), the sequence number and signature generated by the TIM are returned.

Tag	Length (Byte)	Value
C4h	1..32	Tax payer ID → 2.6.6
C5h	1..4	Consecutive number of the TIM → 2.6.7
CBh	1..4	Sequence number of the transaction → 2.6.14
9Eh	48/64	Signature of the transaction → 2.6.21

SW1 / SW2	
90 00h	No error

Response after Negative Plausibility Check or errors

In the event of a negative plausibility check, various result codes are returned, depending on the reason.

SW1 / SW2	
xx xxh	See error messages (→ 4) and ISO 7816-4
62 00h	Warning: command could not be completed
67 00h	LC invalid
6A 86h	P1 / P2 invalid
98 13h	TIM_ERROR_CURRENCY → 4
98 21h	TIM_ERROR_TAX_VERIFICATION_FAILED → 4
98 22h	TIM_ERROR_NEGATIVE_TURNOVER → 4

Turnover plausibility Check of the TIM

The following conditions are checked:

1. The plausibility of turnover, VAT and VAT rate is checked by the TIM. If the flag "VAT not included" was not transmitted, the turnover is regarded as a gross value, otherwise as a net value. If this plausibility check fails, the result code 98 21h TIM_ERROR_TAX_VERIFICATION_FAILED is sent back.
2. If the turnover is negative, a negative turnover must be transmitted to the TIM that is larger than or equal to the absolute amount of the turnover. The turnover and corre-

[16] The TIM carries out a semantic, syntactic and content-based plausibility check for the transmitted data structure and turnover data.

sponding VAT amount are reduced by the respective share of the negative turnover before transmission to the TIM. (Example: 100 Euro sales - 10 Euro goods return = 90 Euro turnover and 10 Euro negative turnover). Negative turnovers are transmitted as positive numbers. If this plausibility check fails, the result code 98 22h TIM_ERROR_NEGATIVE_TURNOVER is sent back.

3. Positive third party turnover is validated against the positive turnover, while negative third party turnover is validated against the negative turnover. In both cases, the amount of the third party turnover must be smaller or the same as the comparative turnover value. If it is larger, the result code 98 23h TIM_ERROR_THIRD_PARTY is sent back.
4. Transaction requests that only include the VAT rate Ex/DBh (x=1..6) (null turnover) are rejected by the TIM.
   Null turnovers in a VAT class may not be transmitted to the TIM.

Only when all plausibility checks have been carried out successfully for the transmitted containers will the transaction be signed and the turnover memories of the TIM updated (see also → 8.2

If a transaction request without turnover is sent, i.e. none of the containers 1..6 are transmitted, because either the resulting turnover for the transaction items in a VAT class gives the value NULL or no turnover is contained in the transaction items, the signature is formed for the other data elements transmitted. A turnover container containing only the field VAT rate would thus be perfectly valid!

Note: In version T.1.1.0 of the TIM, this was answered with the error TIM_ERROR_DATA_MISSING. A transaction without sales (container 1..6 and negative turnover empty, i.e. not transmitted) is useful, for example, to document that goods have been delivered free of charge.

3.4.2 TR Data

If using the command TR Data to sign a freely selectable data block, the transmitted hash value must be caculated over the data block to be signed. The TIM checks the correct length of the transmitted hash value, generates the transaction sequence number, prepares the data in accordance with the hash specification (→Table 7-2), signs the data set and sends back the sequence number and the signature. In the case of a negative result, an error code (→ 4) is sent back.

Command

CLA	INS	P1	P2
80h	40h	01h	00h

LC	Data	LE
*	Transaction data→ Table 3-11	00h

* Length of the transmitted transaction data

Data

The transaction data contains TLV objects according to Table 3-11.

Table 3-11: TR Data Request

Tag	Length (Byte)	Value
C7h	20/32	Hash value of the data to be signed

Response after Positive Plausibility Check

Tag	Length (Byte)	Value
CBh	1..4	Sequence number of the transaction→ 2.6.14
9Eh	48	Signature of the transaction → 2.6.21

Response after Negative Plausibility Check

In the event of a negative plausibility check, various result codes are sent back, depending on the reason.

SW1 / SW2	
xx xxh	See error messages (→4) and ISO 7816-4
67 00h	LC invalid
6A 86h	P1 / P2 invalid

3.4.3 TR _Tax Payer

The command TR TaxPayer largely corresponds to the command TR Data, see 3.4.2 in relation to the TR Tax Payer request. However, the response also contains the tags C4h and C5h, which are added to the signature by the TIM in accordance with the hash regulation Table 7-2. In the case of a negative result, an error code (→4) is sent back.

Command

CLA	INS	P1	P2
80h	40h	02h	00h

LC	Data	LE
*	Transaction data → Table 3-12	00h

* Length of the transmitted hash value

Data

The transaction data contains TLV objects according to Table 3-12.

Table 3-12: TR Tax Payer Request

Tag	Length (Byte)	Value
C7h	20/32	Hash value of the data to be signed

Response after Positive Plausibility Check

Tag	Length (Byte)	Value
C4h	1..32	Tax payer ID → 2.6.6
C5h	1..4	Sequence number of the TIM → 2.6.7
CBh	1..4	Sequence number of the transaction → 2.6.14
9Eh	48	Signature of the transaction → 2.6.21

Response after Negative Plausibility Check

See 3.4.2.

3.4.4 TR Time Stamp

The command TR Time Stamp largely corresponds to the command TR Data, see 3.4.2. In addition to the hash value for the data block, the request TR Time Stamp must, however, also contain the tags for the date and time. The response is identical to the response in 3.4.3. In the case of a negative result, an error code (→4) is sent back.

Command

CLA	INS	P1	P2
80h	40h	03h	00h

LC	Data	LE
*	Transaction data→ Table 3-13	00h

* Length of the transmitted hash value

Data

The transaction data contains TLV objects according to Table 3-13.

Table 3-13: TR Time Stamp Request

Tag	Length (Byte)	Value
CDh	4	Date → 2.6.15
CEh	2	Time → 2.6.16
C7h	20/32	Hash value of the data to be signed

Response after Positive Plausibility Check

Tag	Length (Byte)	Value
C4h	1..32	Tax payer ID → 2.6.6
C5h	1..4	Sequence number of the TIM → 2.6.7
CBh	1..4	Sequence number of the transaction → 2.6.14
9Eh	48	Signature of the transaction → 2.6.21

Response after Negative Plausibility Check

See 3.4.2.

3.5 REPORT (Daily Closing)

The REPORT command outputs the totalising memory of the TIM. The command offers five variants that are distinguished by the encoding of parameter P1. The variants are explained in detail in the following sections.

The content of the data field depends on the command variant. The length of the expected response has to be encoded in the LE field (00h → all data).

Command

CLA	INS	P1	P2
80h	42h	01h / 02h / 03h / 04h / 05h	00h / 01h / 02h

LC	Data	LE
*	Data (see → 3.5.1 - 3.5.5)	00h

* Length of the transmitted data

Parameter P1

P1	Meaning
01h	REPORT Signed
02h	REPORT Unsigned
03h	REPORT Span
04h	REPORT TIM Activate
05h	REPORT TIM Deactivate

Response

The response is identical for each of these report commands and is therefore shown only once.

Table 3-14: REPORT Response

Tag	Length (Byte)	Value	possi-bly not trans-mitted[17]
C0h	1	Life cycle of the TIM → 2.6.2	
C4h	1..32	Tax payer ID → 2.6.6	
C5h	1..4	Consecutive number of the TIM → 2.6.7	
CCh	1..4	Sequence number of the report → 2.6.14	
D2h	1..4	Sequence number of the first transaction of the turnover period → 2.6.14	
D3h	1..4	Sequence number of the last transaction of the turnover period → 2.6.14	
E1h	6..34	Container 1 → 2.6.21	X
D8h	1..11	Turnover → 2.6.18	
D9h	1..11	Negative turnover → 2.6.18	X
DBh	1..2	VAT rate → 2.6.19	
DDh	0	Flag Turnover overflow → 2.6.13	X
DEh	0	Flag for Change in VAT rate → 2.6.13	X
		…	X
E6h	6..34	Container 6 → 2.6.21	X
D8h	1..11	Turnover → 2.6.18	
D9h	1..11	Negative turnover → 2.6.18	X
DBh	1..2	VAT rate → 2.6.19	
DDh	0	Flag Turnover overflow → 2.6.13	X
DEh	0	Flag for Change in VAT rate → 2.6.13	X
E7h	6..23	Container third-party → 2.6.21	X
D8h	1..13	Turnover → 2.6.18	
DCh	1..4	Transaction counter → 2.6.20	
DDh	0	Flag Turnover overflow → 2.6.13	X
E8h	6..23	Container delivery note → 2.6.21	X
D8h	1..11	Turnover → 2.6.18	

[17] Non-booked containers and flags not set are not transmitted.

Tag	Length (Byte)	Value	possibly not transmitted[17]
DCh	1..4	Transaction counter → 2.6.20	
DDh	0	Flag Turnover overflow → 2.6.13	X
E9h	6..21	Container training → 2.6.21	X
D8h	1..11	Turnover → 2.6.18	
DCh	1..4	Transaction counter → 2.6.20	
DDh	0	Flag Turnover overflow → 2.6.13	X
9Eh	48/64	Signature → 2.6.1	X[18]

For each container, the VAT rate returned in each case is the VAT rate of the last month read out.

SW1 / SW2	
xx xxh	See error messages (→ 4) and ISO 7816-4
67 00h	LC invalid
6A 80h	Invalid parameter in data field
98 D1h	Answer needs “extendend length”
90 00h	No error

Remarks to error codes:

The response to the REPORT command can exceed the length of 256 bytes. In this case, the TIM only returns the TIM_WARNING_ANSWER_LENGTH (98 D1h) warning but no REPORT data. In this case, the REPORT command must be repeated as extended length APDU.

3.5.1 REPORT Signed

This command generates a signed report. For this, the totals of all booked turnovers are calculated and sent back. Optionally, a hash value calculated from additional data can be added to the command and therefore be included in the signature.

A signature is generated in accordance with the Hash Specification Report (→ 7.2) and the sequence number of the report is incremented. The response contains the data object Signature (9Eh).

The data field of this command contains date and time in accordance with →Table 3-15. The length of the expected response has to be encoded in the LE field (00h → all data).

[18] The signature is only given back with the commands REPORT Signed, TIM Activate and TIM Deactivate

Command

CLA	INS	P1	P2
80h	42h	01h	00h

LC	Data	LE
*	Date/time/hash value → Table 3-15	00h

* Length of the transmitted data

Data

Data and time and (optional) hash value are indicated as follows:

Table 3-15: Date/Time of the Report Request

Tag	Length (Byte)	Value	possibly not transmitted
CDh	4	Date → 2.6.15	
CEh	2	Time → 2.6.16	
D4h	20/32	Hash value of the report items	X

Response

See → 3.5

3.5.2 REPORT Unsigned

This command generates an unsigned report. For this, the totals of all booked turnovers are calculated and sent back.

No signature is generated. The sequence number of the report remains unchanged. No data object Signature (9Eh) is transmitted in the response.

The data field of this command contains date and time in accordance with → Table 3-16: Date/Time of the Report Request

.

The length of the expected response has to be encoded in the LE field (00h → all data).

Command

CLA	INS	P1	P2
80h	42h	02h	00h / 01h / 02h

LC	Data	LE
0Ah	Date/time →Table 3-16	00h

Parameter P2

P2	Meaning
00h	Memory errors are reported (98 E2h).
01h	Memory errors will be ignored. Monthly turnovers with errors are added up
02h	Memory errors will be ignored. Monthly turnovers with errors are **not** added up

Data

Data and time are indicated as follows:

Table 3-16: Date/Time of the Report Request

Tag	Length (Byte)	Value
CDh	4	Date → 2.6.15
CEh	2	Time → 2.6.16

Response

See → 3.5

3.5.3 REPORT Span

This command calculates the turnover totals over a specific date range (from / to).

Starting month and finishing month for the summation are specified in the data field of this command. The length of the expected response has to be encoded in the LE field (00h → means all data). The turnover data are not signed and the sequence number of the report remains unchanged. The data object Signature (9Eh) is not transmitted in the response.

Command

CLA	INS	P1	P2
80h	42h	03h	00h./ 01h / 02h

LC	Data	LE
14h	**Turnover period → Table 3-17**	00h

Parameter P2

P2	Meaning
00h	Memory errors are reported (98 E2h).
01h	Memory errors will be ignored. Monthly turnovers with errors are added up
02h	Memory errors will be ignored. Monthly turnovers with errors are **not** added up

Data

The turnover period is indicated as follows:

Table 3-17: Turnover Period

Tag	Length (Byte)	Value
CDh	4	Date → 2.6.15
CEh	2	Time → 2.6.16
D0h	3	First month of the turnover period → 2.6.15
D1h	3	Last month of the turnover period → 2.6.15

Response

See → 3.5

3.5.4 REPORT TIM Activate

This command activates the TIM. This triggers the transition of the TIM to LC 03, see 5.

For this, the turnover totals are summarised over all the months and returned together with a sequence number and signature. The TIM transport PIN, the date and the time must be transmitted in the data field of this command. The length of the expected response has to be encoded in the LE field (00h → all data).

Command

CLA	INS	P1	P2
80h	42h	04h	00h

LC	Data	LE
12h	Activation data → Table 3-18	00h

Data

Table 3-18: Activation Data

Tag	Length (Byte)	Value
CDh	4	Date → 2.6.15
CEh	2	Time → 2.6.16
C3h	6	Transport PIN → 2.6.5

Response

See → 3.5

3.5.5 REPORT TIM Deactivate

This command permanently deactivates the TIM! Renewed activation is not possible (see Life cycle of the TIM → 5).

For this, the turnover totals are summarised over all the months and returned together with a sequence number and signature.

The length of the expected response has to be encoded in the LE field (00h → all data).

Command

CLA	INS	P1	P2
80h	42h	05h	00h

LC	Data	LE
0Ah	Deactivation data → Table 3-19	00h

Data

Table 3-19: Deactivation data

Tag	Length (Byte)	Value
CDh	4	Date → 2.6.15
CEh	2	Time → 2.6.16

Response

See → 3.5

3.6 GET LATEST RESPONSE

The command GET LATEST RESPONSE can be used to call up the results of the last TRANSACTION or REPORT command again. The results of a TRANSACTION or REPORT command are saved separately from one another in non-volatile memory; the results can thus be read even after a power loss or after resetting the card. Use of this command is optional.

If the response to a TRANSACTION or REPORT command is not properly received due to a communication error between the card and system, the system can call up the response again using GET LATEST RESPONSE.

If the expected sequence number is not supplied here, the previous TRANSACTION or REPORT command was not properly completed. In this case, the relevant TRANSACTION or REPORT command needs to be performed again.

The parameter P1 selects the memory areas:

P1 = 00h Result of the last TRANSACTION
P1 = 01h Result of the last REPORT

Command

CLA	INS	P1	P2
80h	C0h	00h/01h	00h

LC	Data	LE
--	--	00h

Response

See 3.4, if GET LATEST RESPONSE has been successfully completed.

If GET LATEST RESPONSE is not successful, the following error message is generated:

SW1 / SW2	
98 E2h	Checksum error in the LATEST REPONSE memory. LATEST RESPONSE has been deleted.
65 00h	Execution error, status of the non-volatile memory possibly changed, no information can be delivered

3.7 VERIFY SIGNATURE

The command VERIFY SIGNATURE allows the signature received for all variants of the TRANSACTION command to be verified. Use of all variants of this command is optional.

The data for the signature verification are sorted by the TIM after the transmission in accordance with the hash specification TRANSACTION (→ 7.1) and then verified using the public key. The signature of a report (REPORT) cannot be verified with this command.

The length of the transmitted data is indicated in the LC field (marked with * below). The length of the expected response has to be encoded in the LE field (00h → all data).

Command

CLA	INS	P1	P2
80h	44h	00h, 01h / 02h / 03h	00h

LC	Data	LE
*	Data for the signature verification → dependent on the variant	00h

* Length of the transmitted data

Parameter P1

P1	Meaning
00h	TRANSACTION
01h	TR Data
02h	TR Tax Payer
03h	TR Time Stamp

3.7.1 TRANSACTION

Command

CLA	INS	P1	P2
80h	44h	00h	00h

LC	Data	LE
*	Transaction data → Table 3-20	00h

* Length of the transmitted transaction data

Data

The original data of the transaction, the sequence number received and the signature received are transmitted for verification of the signature received. The TLV objects according toTable 3-20 must be transmitted.

Table 3-20: Data for the Signature verification TRANSACTION

Tag	Length (Byte)	Value	possibly not transmitted[19]
CDh	4	Date → 2.6.15	
CEh	2	Time → 2.6.16	
C6h	1..16	Operator ID → 2.6.8	
C7h	20/32	Hash value of the item data → 2.6.9	
C9h	*0*	*Flag "VAT not included" → 2.6.13*	*X*
CAh	*0*	*Flag "Training mode" → 2.6.13*	*X*
D7h	0	Flag “Delivery note” → 2.6.13	X
CBh	1..4	Sequence number of the transaction → 2.6.14	
E1h	6..28	Container 1 → 2.6.21	X
D8h	1..6	Turnover → 2.6.18	X
D9h	1..6	Negative turnover → 2.6.18	X
DAh	1..6	Value-added tax → 2.6.18	X
DBh	1..2	VAT rate → 2.6.19	
D6h	1..6	Third party turnover → 2.6.18	X
		…	X
E6h	6..28	Container 6 → 2.6.21	X
D8h	1..6	Turnover → 2.6.18	X
D9h	1..6	Negative turnover →2.6.18	X
DAh	1..6	Value-added tax → 2.6.18	X
DBh	1..2	VAT rate → 2.6.19	
D6h	1..6	Third party turnover → 2.6.18	X
9Eh	48/64	Signature of the transaction → 2.6.1	

Response

In the case of a positive signature verification, SW1/SW2 = 90 00h is returned. If the signature verification fails, SW1/SW2 = 98 31h is returned. Further error codes are possible.

[19] Null turnovers and flags not set are not transmitted.

SW1 / SW2	
xx xxh	See error messages (→ 4) and ISO 7816-4
67 00h	LC invalid
6A 80h	Invalid parameter in data field
98 31h	TIM_ERROR _SIGNATURE_INVALID → 4
90 00h	No error

3.7.2 TR Data

Command

CLA	INS	P1	P2
80h	44h	01h	00h

LC	Data	LE
*	Transaction data → Table 3-21	00h

* Length of the transmitted transaction data

Data

The hash value of the signed data set, the received sequence number and the received signature are transmitted for verification of the received signature. The TLV objects according to Table 3-21 must be transmitted.

Table 3-21: Data for the Signature Verification (TR Data)

Tag	Length (Byte)	Value	possibly not transmitted
C7h	20/32	Hash value of the data to be signed	
CBh	1..4	Sequence number of the transaction → 2.6.14	
9Eh	48/64	Signature of the transaction → 2.6.1	

Response

See 3.7.1

3.7.3 TR Tax Payer

Command

CLA	INS	P1	P2
80h	44h	02h	00h

LC	Data	LE
*	Transaction data → Table 3-22	00h

* Length of the transmitted transaction data

Data

The hash value of the signed data set, the received sequence number and the received signature are transmitted for verification of the received signature. The TLV objects according to Table 3-22 must be transmitted.

Table 3-22: Data for the Signature Verification TR Tax Payer)

Tag	Length (Byte)	Value	possibly not transmitted
C7h	20/32	Hash value of the data to be signed	
CBh	1..4	Sequence number of the transaction → 2.6.14	
9Eh	48/64	Signature of the transaction → 2.6.1	

Note: Although the data transmitted for the signature verification is identical to the data from TR Data, the TIM incorporates the internally saved information with the tags C4h and C5h into the calculation.

Response

See 3.7.1

3.7.4 TR Time Stamp

Command

CLA	INS	P1	P2
80h	44h	03h	00h

LC	Data	LE
*	Transaction data → Table 3-23	00h

* Length of the transmitted transaction data

Data

The date, time and hash value of the signed data set, the received sequence number and the received signature are transmitted for verification of the received signature. The TLV objects according to Table 3-23 must be transmitted.

Table 3-23: Data for the Signature Verification TR Time Stamp

Tag	Length (Byte)	Value	possibly not trans-mitted
CDh	4	Date → 2.6.15	X
CEh	2	Time → 2.6.16	X
C7h	20/32	Hash value of the data to be signed	
CBh	1..4	Sequence number of the transaction → 2.6.14	
9Eh	48/64	Signature of the transaction → 2.6.1	

Note: The TIM also incorporates the internally saved information for the tags C4h and C5h into the calculations for the verification.

Response

See 3.7.1

3.8 HASH

Depending on the selected cryptographic algorithm when the order is placed, the command HASH allows a SHA-1 hash value to be calculated with the TIM T.1.1.0 or a SHA-256 hash value to be calculated with the TIM V.2.1.0. Use of this command is optional.

The HASH command can be used, for example, to use the TIM for calculation of the hash value of the transaction items (→ 2.6.9). The exact specifications for this can be found in the definition of the respective profile (→ 9). The transaction items are transmitted to the TIM in accordance with the hash specification for transaction items. The TIM calculates the hash value and sends it back. The TIM does not perform any verification of the data.

In most cases, use of the HASH command is not optimal for efficient implementation. Instead the calculation of the hash value for the transaction items should be carried out on the host. During the development, the HASH command can be used to verify your own SHA-1 or SHA-256 implementation.

The HASH command is implemented with the ISO 7816-8 command PSO_HASH (PSO_H). The encoding of the parameters P1 and P2 corresponds to the ISO standard.

The data can be transmitted in several steps ("chaining"). The CLA byte 10h is used to indicate that further data blocks are to follow. Except in the last step (CLA = 00h), the data must be transmitted in multiples of 64 bytes (40h).

The length of the transmitted data is indicated in the LC field. The length of the expected response has to be encoded in the LE field (00h → all data or 14h → length of an SHA-1 hash value or 20h → length of a SHA-256 hash value).

Command

CLA	INS	P1	P2
00h / 10h	2Ah	90h	80h

LC	Data	LE
*	Data	00h / 14h or 20h

* Length of the transmitted data

Data

Data for which the SHA-1 or SHA-256 hash value is to be calculated.

Response

Data
Hash value of the data (20/32 bytes) / blank (0 bytes)

The hash value is returned only with the last command (CLA = 00h); for all other calls (CLA = 10h) the return data are empty.

SW1 / SW2	
xx xxh	See error messages (→ 4) and ISO 7816-8
90 00h	No error

4 Error Messages (RESULT CODES)

Error messages of the TIM application are transmitted via result codes of the commands. These codes are transmitted in the response bytes SW1 and SW2 (→ ISO 7816-4).

Further ISO 7816-4 error codes can occur in addition to these application-specific codes.

Table 4-1 contains the error codes that are returned by the TIM application:

Table 4-1: Error messages of the TIM

Encoding SW1/SW2	Name	Description
90 00h	NO_ERROR	Command successfully executed. (ISO 7816-4)
98 01h	TIM_ERROR_TLV	Error in TLV format (length, invalid tags, etc.)
98 02h	TIM_ERROR_VALUE	The payload data field (Value) of at least one TLV object does not correspond to the specification (e.g. null turnover, sign encoding not correct) With composite data objects, an error in the simple data objects contained is signalled in this way.
98 03h	TIM_ERROR_ DATA_MISSING	The transmitted data are not correct (mandatory data object missing in the transmitted data, etc.).
98 04h	TIM_ERROR_ INVALID_CHARACTER	The payload data field (Value) of a TLV object contains at least one invalid character (e.g. in operator ID).
98 11h	TIM_ERROR_ DATE_FORMAT	The date or time is not plausible (e.g. 32.12.2010 or 24:00 h).
98 12h	TIM_ERROR_ DATE_OUT_OF_RANGE	The transmitted date lies outside the period of validity of the TIM.
98 13h	TIM_ERROR_ CURRENCY	The transmitted currency code does not correspond to the code stored on the TIM.
98 21h	TIM_ERROR_ TAX_VERIFICATION_FAILED	Verification of the transmitted values for tax, turnover and VAT rate failed.
98 22h	TIM_ERROR_ NEGATIVE_TURNOVER	The transmitted negative turnover is not plausible.
98 23h	TIM_ERROR_THIRD_PARTY	The transmitted third party turnover is larger than the turnover subject to VAT.
98 31h	TIM_ERROR_ INVALID_SIGNATURE	The data transmitted with the command VERIFY SIGNATURE (transaction data, transaction sequence number received and signature received) were not generated by the TIM.
98 41h	TIM_ERROR_ INVALID_LIVECYCLE	The command cannot be executed in the current life cycle of the TIM.
98 D1h	TIM_WARNING_ ANSWER_LENGTH	The length of the response requires an "extended length" APDU. Data on the TIM has not been changed. The command can be repeated.

Encoding SW1/SW2	Name	Description
98 D2h	TIM_WARNING_INTERNAL	An error - not specified - has occurred during processing. Data on the TIM has not been changed. The command can be repeated.
98 E0h	TIM_ERROR_OUT_OF_ MEMORY	Insufficient memory on the smart card.
98 E1h	TIM_ERROR_ MEMORY_FAILURE	Error during writing to EEPROM, error during reading from EEPROM. The TIM was deactivated for security reasons.
98 E2h	TIM_ERROR_ DATA_CORRUPTED	The data stored on the TIM are not correct (checksum error).
98 F2h	TIM_ERROR_INTERNAL	An error - not specified - has occurred during processing. The TIM was deactivated for security reasons
98 FFh	TIM_ERROR_ NOT_SUPPORTED	Function / data field is currently not supported by the TIM
65 00h	TIM_ERROR_ EXECUTION_ERROR	Execution error, state of the non-volatile memory has possibly been changed, no information can be given.

5 Life Cycle of the TIM

5.1 Encoding of the TIM Life Cycle

Table 5-1: Encoding of the TIM life cycle

Name	Encoding	Description
(UNDEFINED)	00h	Undefined TIM life cycle
TIM_INITIALISED	01h	The TIM package was successfully uploaded to the smart card
TIM_PERSONALISED	02h	The TIM is personalised for the tax payer, the TP_ID, TP_ID_NO, currency code, personalisation date and expiry date have been entered.
TIM_ACTIVATED	03h	The TIM has been activated using the transport PIN. Transactions can now be signed, values saved and reports generated.
TIM_DEACTIVATED	04h	The TIM is deactivated. Transactions are no longer signed or saved. The stored data can still be read out.

5.2 Transitions in the Life Cycle of the TIM

Table 5-2: State transitions in the life cycle of the TIM

State	Transition condition	Subsequent state
Smart card without TIM package	Upload of the TIM package	TIM_INITIALISED
TIM_INITIALISED	Personalisation of the TIM by the authorised body	TIM_PERSONALISED
TIM_PERSONALISED	Activation of the TIM by the tax payer using the transport PIN	TIM_ACTIVATED
TIM_PERSONALISED	Incorrect input of the transport PIN 16 times by the tax payer	TIM_DEACTIVATED
TIM_ACTIVATED	Deactivation of the TIM by the tax payer	TIM_DEACTIVATED
TIM_DEACTIVATED		None

The transport PIN is defined as a sequence of ASCII characters and consists of exactly 6 characters. The error counter for the PIN has the initial value 15. The PIN must be verified once with the command REPORT TIM Activate (80h 42h 04h 00h) (→ 3.5.4). The TIM is then activated.

The TIM can be permanently deactivated using the command REPORT TIM Deactivate (80h 42h 05h 00h) (→ 3.5.5). Further transactions are then no longer possible, but read access remains possible.

The TIM is delivered to the tax payer in the life cycle TIM_PERSONALISED. If the transport PIN is not to be used, the TIM can also be delivered in the life cycle TIM_ACTIVATED.

5.3 Available Commands per TIM Life Cycle

Table 5-3 gives an overview of the possible commands in the different life cycles.

Table 5-3: Available commands per TIM life cycle

TIM command	TIM life cycle				
	Smart card without TIM package	TIM_INITIALISED	TIM_PERSONALISED	TIM_ACTIVATED	TIM_DEACTIVATED
SELECT FILE	§	X	X	X	X
GET DATA TIM Status	-	X	X	X	X
GET DATA TIM Status extended	-	- *	X	X	X
GET DATA Booked Months	-	- *	X	X	X
GET DATA Hash Length	-	X	X	X	X
GET DATA Cryptographic Algorithms	-	X	X	X	X
GET DATA Memory Status	-	X	X	X	X
TRANSACTION (all variants)	-	- *	- *	X	- *
REPORT signed	-	- *	- *	X	X
REPORT unsigned	-	- *	- *	X	X
REPORT Span	-	- *	- *	X	X
REPORT TIM Activate	-	- *	X	- *	- *
REPORT TIM Deactivate	-	- *	- *	X	- *
GET LATEST RESPONSE	-	- *	- *	X	X
VERIFY SIGNATURE (all variants)	-	- *	- *	X	X
HASH	§	X	X	X	X
READ CERTIFICATE	§	X	X	X	X

* The RESULT CODE 98 41h - TIM_ERROR_INVALID_LIVECYCLE is sent back

§ ISO7816, depending on the smart card operating system

6 Definitions and Specifications

6.1 VAT classes

6.1.1 Definition of the Containers 1..6

Six different containers are defined for VAT-related values, each of which represents one VAT class. The VAT class is defined here independently on the respective current VAT rate. The specification of VAT rates is laid down by statutory regulations. It is performed in the cash register and is not saved on the TIM in advance. The VAT rate of the last booking to a particular month is, however, stored in the respective container.

Table 6-1: Definition of containers 1..6 in accordance with the VAT classes

Name	**Tag**	**Designation of VAT class**	Designation and VAT rate in Germany (2016)
TIM_CONTAINER_VAT_1	E1h	Standard	Standard rate: 19%
TIM_CONTAINER_VAT_2	E2h	Reduced 1	Reduced rate 7%
TIM_CONTAINER_VAT_3	E3h	Reduced 2	(does not exist)
TIM_CONTAINER_VAT_4	E4h	VAT free	0%
TIM_CONTAINER_VAT_5	E5h	Special 1	Average rate 1: 10.7%
TIM_CONTAINER_VAT_6	E6h	Special 2	Average rate 2: 5.5%

This arrangement should allow the majority of VAT models to be implemented. Within the European Union this will probably comply to all models . For the arrangement, see: Official Journal of the European Union L347: "Council Directive 2006/112/EC of 28 November 2006 on the common system of value added tax" under:
http://eur-lex.europa.eu/JOHtml.do?uri=OJ%3AL%3A2006%3A347%3ASOM%3ADE%3AHTML

6.2 Character Substitution

In order to be able to verify printed documents, characters must be substituted in the payload data of the following TLV objects by the system issuing the commands:

- TIM operator ID – C6h (→ 2.6.8),
- Text fields defined in the respective profile.

This applies to characters that cannot be printed or which cannot be unambiguously recovered from the printed image. The character substitution thus forms the basis for verification of the hash value of the transaction items on the document.

The character substitution has to be performed during a transaction before the TLV object TIM operator ID is transmitted to the TIM.

This requires to perform the following steps (in the order specified here):

If the printing length on the sales receipt is less than 16 characters:
Shorten the character chain to printing length

Non-printable characters and blanks (ASCII characters < 0x21 and = 0x7F) are omitted

Substitution of: ABCDEFGHIJKLNOPQRSTUVWXYZ
by: abcdefghijklnopqrstuvwxyz

Permitted characters: #0123456789abcdefghijklmnopqrstuvwxyz

All other characters are replaced by the substitution character "**#**"

The character string is cut off after 16 characters

Before the data sets are stored in the XML export file, at point 1 must have been performed (see document → INSIKA Export Format). The further steps should be processed during verification to keep the information in the export file as intelligible as possible.

The character substitution is designed such that only the first run changes the data. Each further run does not make changes to the data.

Further information about data requiring character substitution is included in the documentation of the profiles.

6.3 Rounding

Rounding is performed according to commercial principles to integer values of the smallest currency unit (here: 1 Euro Cent) (see also DIN 1333, Feb. 1992).

0.5 of the smallest currency unit is added to a positive number and in the result the digits behind the decimal point are omitted. With a negative number the amount is rounded according to the principle described above. The minus sign is placed in front of the rounded amount.

7 Information on Signature Verification

The information presented in this chapter allows the signature verification to be carried out. This information is initially not necessary for basic integration for the signature generation with the TIM.

7.1 Hash Specification TRANSACTION

For calculation of the signatures for transactions and freely selectable data sets, the transmitted data objects are first arranged on the TIM in the order shown in Table 7-1 or Table 7-2. As the first step in the ECDSA process, a SHA-1/SHA-256 hash value, see footnote 1, is formed for these data on the TIM that is then signed.

7.1.1 TRANSACTION

The data objects in Table 7-1 are signature-relevant, i.e. they go directly into the signature. All data objects are encoded as TLV (→ 2.1).

Table 7-1: Hash specification TRANSACTION

Tag	Length (Byte)	Value	possibly not transmitted[20]	Sequence
CDh	4	Date → 2.6.15		1
CEh	2	Time → 2.6.16		2
C4h	1..32	Tax payer ID → 2.6.6		3
C5h	1..4	Consecutive number of the TIM → 2.6.7		4
C6h	1..16	Operator identification → 2.6.8		5
C7h	20/32	Hash value of the item data → 2.6.9		6
C9h	0	Flag: "VAT not included" → 2.6.13	X	7
CAh	0	Flag: "Training mode" → 2.6.12	X	8
D7h	0	Flag “Delivery note” → 2.6.13	X	0
CBh	1..4	Sequence number of the transaction → 2.6.14		9
E1h	6..12	Container 1 → 2.6.21	X	10 .. 39[21]
D8h	1..6	Turnover → 2.6.18		
DBh	1..2	VAT rate → 2.6.19		
D6h	1..6	Third party turnover → 2.6.18	X	
			X	

[20] Null turnovers and flags not set are not transmitted.

[21] Order by ascending container numbers 1-6, non-booked containers do not go into the hash

Tag	Length (Byte)	Value	possibly not trans-mitted[20]	Sequence
		…		
E6h	6..12	Container 6 → 2.6.21	X	
D8h	1..6	Turnover → 2.6.18		
DBh	1..2	VAT rate → 2.6.19		
D6h	1..6	Third party turnover → 2.6.18	X	

The hash value is formed for the turnovers and flags transmitted to the command. Only the data objects "Turnover" (D8h), "VAT rate" (DBh) and “Third party turnover” (D6h) are used in composite data objects, not the data objects "Negative turnover" (D9h) and "VAT rate" (DAh).

If the field “Turnover” (D8h) is not transmitted in a turnover container, it is nevertheless taken into account in the formation of the hash value. In this case, the value 0 with a minimum length is encoded (D8 01 0C) and hashed. Other turnovers or flags not transmitted are not added by the TIM.

7.1.2 TR Data, TR Tax Payer, TR Time Stamp

The data objects in Table 7-2 are signature-relevant depending on the command variant, i.e. they go directly into the signature. All data objects are encoded as TLV (→ 2.1).

Table 7-2: Hash Regulations TR Data, TR Tax Payer, TR Time Stamp

Tag	Length (Byte)	Value	possibly not trans-mitted	Sequence
CDh	4	Date → 2.6.15	X	1
CEh	2	Time → 2.6.16	X	2
C4h	1..32	Tax payer ID → 2.6.6	X	3
C5h	1..4	Sequence number of the TIM → 2.6.7	X	4
C7h	20/32	Hash value of the data to be signed		5
CBh	1..4	Sequence number of the transaction → 2.6.14		6

7.2 Hash Specification REPORT

For calculation of the signatures for reports, the TLV objects sent back with the command REPORT Signed, REPORT TIM Activate or REPORT TIM Deactivate are hashed as shown in Table 7-3 and then signed.

The fields "Turnover" (D8h) and "Negative turnover" (D9h) are always hashed if a turnover container has been booked, even if the relevant turnover total gives the value 0. A null turnover is coded with the minimum length (Dx 01 0C) and hashed.

Table 7-3: Hash Regulation REPORT

Tag	Length (Byte)	Value	possibly not transmitted[22]	Sequence
CDh	4	Date → 2.6.15		1
CEh	2	Time → 2.6.16		2
D4h	20/32	TIM hash value of the report items	X	3
C0h	1	TIM Life Cycle → 2.6.2		4
C4h	1..32	Tax payer ID → 2.6.6		5
C5h	1..4	Consecutive number of the TIM → 2.6.7		6
CCh	1..4	Sequence number of the report → 2.6.14		7
D2h	1..4	Sequence number of the first transaction → 2.6.14		8
D3h	1..4	Sequence number of the last transaction → 2.6.14		9
E1h	6..34	Container 1 → 2.6.21	X	10..57
D8h	1..11	Turnover → 2.6.18		
D9h	1..11	Negative turnover → 2.6.18	X	
DBh	1..2	VAT rate → 2.6.19		
DDh	0	Flag Turnover overflow → 2.6.12	X	
DEh	0	Flag for Change in VAT rate → 2.6.12	X	
		…	X	
E6h	6..34	Container 6 → 2.6.21	X	
D8h	1..11	Turnover → 2.6.18		
D9h	1..11	Negative turnover → 2.6.18		
DBh	1..2	VAT rate → 2.6.19		
DDh	0	Flag Turnover overflow → 2.6.12	X	
DEh	0	TIM flag for Change in VAT rate → 2.6.12	X	
E7h	6..23	Container third-party → 2.6.21	X	
D8h	1..11	Turnover → 2.6.18		

[22] Null turnovers and flags not set are not transmitted.

Tag	Length (Byte)	Value	possibly not transmitted[22]	Sequence
DCh	1..4	Transaction counter → 2.6.20		
DDh	0	Flag Turnover overflow → 2.6.12	X	
E8h	6..23	Container delivery note → 2.6.21	X	
D8h	1..11	Turnover → 2.6.18		
DCh	1..4	Transaction counter → 2.6.20		
DDh	0	Flag Turnover overflow → 2.6.12	X	
E9h	6..21	Container training → 2.6.21	X	
D8h	1..11	Turnover → 2.6.18		
DCh	1..4	Transaction counter → 2.6.20		
DDh	0	Flag Turnover overflow → 2.6.12	X	

Flags not set and VAT totalisers not booked are not included in the hash.

Only the TLV objects "Turnover" (D8h), "Negative turnover" (D9h), "VAT rate" (DBh), "Transaction counter" (DCh), "Flag Turnover overrun" (DDh) and "Flag Change in VAT rate" (DEh) are used in composite data objects (tags E1h … E9h), not the TLV object "VAT" (DAh). Turnover totalisers not booked or flags not set are not added by the TIM.

7.3 Hash and Signature Algorithms

The TIM uses only open and standardised hash and signature procedures.

The hash function SHA-1/SHA-256 (Secure Hash Algorithm) used in TIM V.2.1.0, see footnote 1, is standardised e.g. in FIPS 180/ FIPS 180-4 of the NIST. SHA-1 delivers short hash values. The theoretical weaknesses of SHA-1 only apply to large data which can be modified to a large extend without a chance to detect it. This is not the case for data protected by INSIKA. Nevertheless, SHA-256 was defined in TIM V.2.1.0 as the standard hash function for future applications and enhancements.

The signature procedure ECDSA (Elliptic Curve Digital Signature Algorithm) used in TIM V.2.1.0 is standardised in ANSI X9.62 or FIPS 186.

Signed transactions can be verified on the TIM with the command VERIFY SIGNATURE (→ 3.6). The hash specifications (→ 7) allow applications to be created that permit the verification of signed transactions and reports.

Following a TRANSACTION command, the following steps are performed on the TIM: The TLV objects provided by the TIM are added to the transmitted TLV objects. The TLV objects are then arranged and transformed according to the hash specifications (→ 7.1). A hash value is then formed over these objects and subsequently signed using the private key.

Following a REPORT command, the following steps are performed on the TIM: The TLV objects of the request and the internally identified TLV objects are arranged according to the hash specifications (→ 7.2). A hash value is then formed over these objects and subsequently signed using the private key.

7.4 Domain Parameters

The TIM V.2.1.0 uses the following ECDSA domain parameters for ECDSA-192:

```
P:  0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF FFFFFFFF
a:  0xFFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFFFF FFFFFFFC
b:  0x64210519 E59C80E7 0FA7E9AB 72243049 FEB8DEEC C146B9B1
Gx: 0x188DA80E B03090F6 7CBF20EB 43A18800 F4FF0AFD 82FF1012
Gy: 0x07192B95 FFC8DA78 631011ED 6B24CDD5 73F977A1 1E794811
n:  0xFFFFFFFF FFFFFFFF FFFFFFFF 99DEF836 146BC9B1 B4D22831
h:  0x1
```

These domain parameters are defined in various standards where they are known under the name "ANSIp192r1", "prime192v1", "P-192" or "secp192r1" (see ANSI X9.62, FIPS 186, etc.).

In contrast, the domain parameters NIST P-256 (according to NIST 186-4) are used for ECDSA-256:

```
p =  0xFFFFFFFF 00000001 00000000 00000000 00000000 FFFFFFFF FFFFFFFF FFFFFFFF
a =  0xFFFFFFFF 00000001 00000000 00000000 00000000 FFFFFFFF FFFFFFFF FFFFFFFC
b =  0x5AC635D8 AA3A93E7 B3EBBD55 769886BC 651D06B0 CC53B0F6 3BCE3C3E 27D2604B
Gx = 0x6B17D1F2 E12C4247 F8BCE6E5 63A440F2 77037D81 2DEB33A0 F4A13945 D898C296
Gy = 0x4FE342E2 FE1A7F9B 8EE7EB4A 7C0F9E16 2BCE3357 6B315ECE CBB64068 37BF51F5
n =  0xFFFFFFFF 00000000 FFFFFFFF FFFFFFFF BCE6FAAD A7179E84 F3B9CAC2 FC632551
h =  0x01
```

7.5 Format of the Signature

The signature is returned in a sequence of two 192/256 bit binary values r and s, see footnote 1, in direct succession. 48/64 bytes are thus transmitted in the TLV object "TIM signature" (→ 2.6.1).

7.6 Certificate and Public Key

The certificate on the TIM follows the specifications in ITU-T X.509v3. It is defined in an ASN.1 structure and stored on the TIM in binary DER code.

The certificate has the following ASN.1 structure:

```
Certificate ::= SEQUENCE  {
    tbsCertificate       TBSCertificate,
    signatureAlgorithm   AlgorithmIdentifier,
    signatureValue       BIT STRING  }
```

7.6.1 Length of the Certificate

The following example shows how the length information can be read from the certificate. The start of the example certificate reads:

30h 82h 03h 38h 30h 82h 02h 20h A0h …

The first byte "30h" identifies a sequence of the BER/DER class "constructed universal". The "82h" is the initial byte of the length field that is displayed in the next two bytes. The bytes "03h 38h" indicate the length of the following data section. It is therefore 824 bytes long. The whole certificate thus has a length of 828 bytes.

7.6.2 Public Key in the Certificate

The object **TBSCertificate** contained in the certificate has the following ASN.1 structure:

```
TBSCertificate          ::=     SEQUENCE {
        Version                 [0] EXPLICIT Version DEFAULT v1,
        serialNumber            CertificateSerialNumber,
        signature               AlgorithmIdentifier,
        issuer                  Name,
        validity                Validity,
        subject                 Name,
        subjectPublicKeyInfo    SubjectPublicKeyInfo,
        issuerUniqueID          [1] IMPLICIT UniqueIdentifier OPTIONAL,
        subjectUniqueID         [2] IMPLICIT UniqueIdentifier OPTIONAL,
        extensions              [3] EXPLICIT Extensions OPTIONAL
}
```

The certificate contains the public key. The key length is 192/256 bits, see footnote 1. As the public key represents a point on an elliptic curve, it is encoded in a sequence of two consecutive 192/256 bit binary numbers, see footnote 1. The size is thus 48/64 bytes, see footnote 1.

In the certificate, the public key is stored in the object `SubjectPublicKeyInfo` that has the following ASN.1 structure:

```
SubjectPublicKeyInfo ::=    SEQUENCE {
        algorithm               AlgorithmIdentifier,
        subjectPublicKey        BIT STRING
}
```

The object `AlgorithmIdentifier` contained in the certificate has the following ASN.1 structure:

```
AlgorithmIdentifier     ::=     SEQUENCE {
        algorithm               OBJECT IDENTIFIER,
        parameters              ANY DEFINED BY algorithm OPTIONAL
}
```

The OBJECT IDENTIFIER (OID) for ECDSA must generally contain the id-ecPublicKey algorithm identifier according to the definition

id-public-key-type OBJECT IDENTIFIER ::= { ansi-X9.62 2 }

id-ecPublicKey OBJECT IDENTIFIER ::= { id-publicKeyType 1 }

and the namedCurve OBJECT IDENTIFIER.

The OBJECT IDENTIFIER (OID) for the signature algorithm SHA1ECDSA with the domain parameters prime192v1 is: { 1.2.840.10045.3.1.1 }. In the DER encoded form this results in the OID: "06h 08h 2Ah 86h 48h CEh 3Dh 03h 01h 01h" (see ITU-T X.690).

For example, on the TIM V.2.1.0 for SHA1/ECDSA the object `SubjectPublicKeyInfo` is encoded as follows:

```
…30h 49h
        30h 13h
                06h 07h 2Ah 86h 48h CEh 3Dh 02h 01h
                06h 08h 2Ah 86h 48h CEh 3Dh 03h 01h 01h
        03h 32h 00h 04h ..(48 Byte Public Key)..
```

The first byte "30h" identifies a sequence of the BER/DER class CONSTRUCTED UNIVERSAL. The following byte "49h" indicates the length. It is therefore 73 bytes long.

The "30h" again identifies a sequence of the BER/DER class CONSTRUCTED UNIVERSAL with the length "13h". The "06h" indicates an OBJECT IDENTIFIER. The next byte "08h" indicates the length. This is followed by the OID. The next object indicates optional parameters.

The "03h" indicates a BIT STRING of the BER/DER class UNIVERSAL. The following byte "32h" indicates the length. It is therefore 50 bytes long. The public key is preceded by the bytes "00h 04h". The byte "00h" indicates how many unused bits are contained in the last octet of the BIT STRING. The byte "04h" indicates that it is an uncompressed key. The following 48 bytes are the public key.

For SHA256/ECDSA, the namedCurve OBJECT IDENTIFIER with the domain parameter NIST P-256 {1.2.840.10045.3.1.7} is used. In the DER encoded form this gives the OID: "06h 08h 2Ah 86h 48h CEh 3Dh 03h 01h 07h" and see ITU-T X.690).

For example, on the TIM V.2.1.0 for SHA256/ECDSA the object `SubjectPublicKeyInfo` is encoded as follows:

```
30h 59h
          30h 13h
                06h 07h 2Ah 86h 48h CEh 3Dh 02h 01h
                06h 08h 2Ah 86h 48h CEh 3Dh 03h 01h 07h
          03h 42h 00h 04h <64 Byte Public Key>
```

Note: The OID indicates only the signature algorithm and the domain parameters used. In order to read the object SubjectPublicKeyInfo, complete parsing of the X.509 certificate has to be performed using the ASN.1-structure and the ASN.1 objects (using tag and length)! (see also ITU-T X.509v3)

8 Data on the TIM

8.1 Personalisation Data on the TIM

The TIM is personalised for the tax payer by the authorised body. The data according to Table 8-1 are thereby set on the TIM:

Table 8-1: Personalisation data on the TIM

Personalisation data	Name	Tag	Command for reading out
Life cycle of the TIM	TIM_LIFECYCLE	C0h	GET DATA TIM Status / GET DATA TIM Status extended
Transport-PIN *	TIM_TRANSPORT_PIN	C3h	-
Tax payer ID	TIM_TP_ID	C4h	GET DATA TIM Status extended / TRANSACTION / REPORT
Consecutive number of the TIM referred to a TIM_TP_ID	TIM_TP_ID_NO	C5h	GET DATA TIM Status extended / TRANSACTION / REPORT
Currency code	TIM_CURRENCY	C8h	GET DATA TIM Status extended
Personalisation date	(unsigned BCD)	-	-
First month to which turnover can be booked	TIM_DATE	CDh	GET DATA TIM Booked Months
Last month to which turnover can be booked[23]	TIM_DATE	CDh	GET DATA TIM Status extended
Country code	TIM_COUNTRY_CODE	D5h	GET DATA TIM Status extended
Number of bookable months	(binary)	-	-
Certificate	(File EF_CERT on TIM)	-	READ CERTIFICATE

8.2 Totalising Memory Model of the TIM

Totalising memories are maintained on the TIM. In this section, the structure and contents of this totalising memory will be explained, which are shown schematically in Figure 8-1. The description of these structures is intended to simplify the understanding of the data transmitted with the REPORT command.

Each successfully executed TRANSACTION command updates the totalising memories on the TIM. These totalising memories are read out with the REPORT command.

[23] The lifespan of the cards will be defined during the personalisation

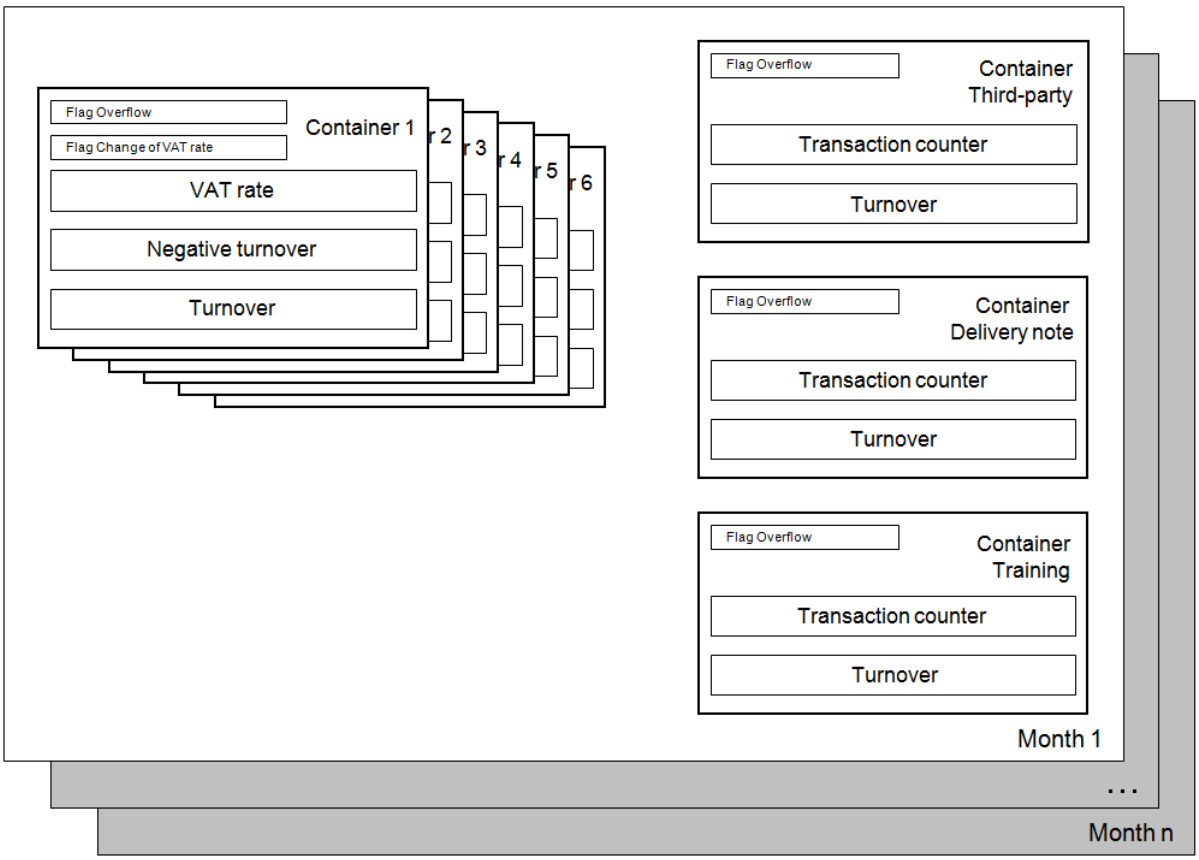

Figure 8-1: Totalising memory model of the TIM

Each month is stored separately on the TIM. The months can be output individually with the command REPORT Span. The number of months is set by the authorised body during the personalisation.

The commands REPORT Signed and REPORT Unsigned return the totals over all the booked months.

The TIM has nine different containers for each month. Containers 1-6 represent the VAT classes (→ 6.1) directly. In addition, the containers Third-party, Delivery note and Training are provided on the TIM. The different containers are explained below.

8.2.1 Containers 1..6

The VAT-related sales from the transactions are totalised in containers 1..6. Containers 1..6 represent the VAT classes (→ 6.1) directly.

The monetary values (turnover and negative turnover) are stored as gross values, i.e. including value-added tax.

With net transactions, the turnovers are converted accordingly by the TIM before addition. In order to reduce the influence of rounding errors here, the values are rounded according to commercial principles with an extended accuracy of 0.0001 of the smallest currency unit in this calculation (→ 6.3). The turnover and negative turnover are stored with this extended accuracy in containers 1..6. The extended accuracy is used only for internal summation and not for output. When reading out the totals using a REPORT command, the values are rounded according to commercial principles to the smallest currency unit.

The field Turnover corresponds to the sales, broken down by tax rates and individual tax exemptions, for the delivery of goods or services. As this involves the total amount of the

transaction, both third-party turnovers and negative turnovers are taken into consideration. Transactions in delivery note or training mode only update the delivery note or training turnover memories.

The VAT rate is entered in the corresponding container with the first transaction of a new month. If a new value for the VAT rate is stored during the current month, the corresponding flag "Change of VAT rate" is set. Changes in the VAT rate are normally only carried out at the end of a month. During a tax audit it is known which VAT rate is valid for which month. The TIM does not check the amount of the transmitted VAT rate, only the calculation of the VAT based on this rate. (see also VAT classes → 6.1)

Negative turnovers in transactions are transmitted in the turnover (100 Euro booked, 10 Euro goods return = 90 Euro turnover). The negative turnovers are transmitted to the TIM in the respective VAT-specific container. Negative turnovers are transmitted as positive values.

During a transaction, the TIM performs a plausibility check of the turnover of the transaction (total VAT-related sales from the transaction) with respect to the negative turnover of the transaction (total VAT-related sales from the negative transaction items). With a negative turnover, the negative turnover must be equal to or larger than the absolute amount of the turnover. This ensures that reduced turnovers due to negative total sums are detected even in the event of a loss of the journal.

8.2.2 Container Third-party E7h

Third-party turnovers are sales of goods or services on behalf of and for the account of a third party. A separate container is created for each month.

Third-party turnovers are included in the turnovers for the containers in the associated VAT classes and are consequently also signed with the turnovers. Third-party turnovers are transmitted to the TIM in container E7h in the container for the corresponding VAT classes with the command TRANSACTION for updating of the third-party totals and are thus included in the signature of the transaction (cf.→ 7.1). If the third-party total is not correctly transmitted to the TIM by the cash register, the tax payer cannot gain any benefit as the turnovers are then regarded as regular turnover.

Third-party turnovers are generally subject to a "four-eyes" principle. Here two parties have an interest in the integrity of the data. The check is generally performed in downline systems. For plausibility checks, the number of transactions in which third-party turnovers occurred is recorded in the "Third-party transaction counter".

8.2.3 Container Delivery Notes E8h

Delivery notes are business transactions where the delivery or service provision needs to be documented but the invoice will be issued at a later point in time. Therefore, they do not represent taxable turnover so that they arealso not added to the turnovers in the containers for the associated VAT classes on the TIM. Consequently, they are also not signed together with these turnovers for a REPORT.

Delivery notes must always be processed as separate transactions. It is not possible to combine them with taxable turnovers in one transaction. They can be saved either with or without price information, in the latter case the turnover container will only save the number of signed delivery notes. As invoicing is carried out in a downline system in which the taxable income can still be influenced by various parameters (discounts, etc.), this sum has only an informative nature.

The delivery note data with the command TRANSACTION is transmitted to the TIM with the flag delivery note. If price information should be transmitted, this is carried out via the container for the relevant VAT class. The turnovers in the containers E1h to E6h that are transmitted for the transaction request are added together internally within the TIM as gross turnover in the container delivery note (E8h). The turnovers are not added to the totalising memory for the turnover container “Turnover 1 to 6”. If the flags delivery note and training are set at the same time, the turnover is booked as training.

8.2.4 Container Training E9h

For transaction requests labelled as training, see 3.4.1, the turnovers transmitted in E1h to E6h are not added to Container 1..6 but are added together to form a total in the totalising memory for training turnover. As these are bookings that do not relate to a transaction, and hence no delivery of goods or other services is involved, the turnovers are not split into VAT rates. Since there is a notable risk or manipulation potential particularly in these bookings, the turnovers are included in the signature of the booking. For plausibility checks, the number of training bookings is also recorded in the "Training booking counter".

A separate container is created for each month.

9 Profiles

By using profiles INSIKA can used for various applications. Their differences mainly lie in:

- the content and structure of the transaction items,
- any additional data for reports (daily closing) and
- the mechanism to verify the correct usage of the system.

The audit of data should be profile-specific - this aspect is not covered in this document.

The interface to the TIM is independent of the profile, which always is the same. In each application exactly one profile is used. Changing the profile during operation does not make sense and is therefore not supported.

9.1 Transaction Items

The profile defines the data objects of transaction items over which a hash value is calculated. This hash value (→ 2.6.9) is passed to the TIM and signed as part of a transaction (→ 3.4). Thus, these data objects become part of the signature indirectly. With this mechanism a large number of data objects can be signed without the need to transmit them through the TIM interface.

The main task of transaction items is traceability of the various total values that are passed to the TIM.

TLV is the most reasonable way to encode transaction items, because this representation is standardized and the necessary mechanisms are already in place for communication with the TIM. Alternative encoding techniques, however, are not ruled out.

9.2 End of Day Data

In the simplest case the contents of a report (→ 3.5) are fully supplied by the TIM (based on the totaliser memory in the TIM). However, it may be useful to record additional information and to protect them by including them into the signature.

To achieve this, the same mechanism as for the transaction items can be used. Again, a hash value over the additional information is calculated (→ 2.6.10) following the rules defined in the profile. This value is transferred to the TIM.

9.3 Control Mechanism

For each application of INSIKA and thus for each profile it has to be defined how the correct usage of the system can be checked. Usually the most reasonable way is issuing signed, verifiable receipts - these may be issued on paper or electronically (stored on a server).

If possible, a check should be possible entirely based on the receipts and access to the cryptographic certificates. For this purpose, the receipt must meet the following requirements:

- It must be possible to reconstruct all transactions from the receipt so that they exactly match the data at the time of the signature (content, form and order). So both hash value are identical.
- All data that are not included in the hash value of the transaction items but are transferred to the TIM (e.g. date and time) or that are returned by the TIM (e.g. the taxpayer ID and signature) must be clearly intelligible on the receipt.

- For a simplified verification in individual cases (useful for printed receipts) the calculation of the hash over the transaction items can be omitted. So the transaction items are not verified but only totals, date, time etc. Since the hash value is necessary for this type of verification of the signature, it must be apparent from the receipt.
- A (largely) automatic verification of receipts is the ideal solution. With electronic documents this is easily possible – for printed documents the most obvious choice is encoding the required data in a machine-readable code (e.g. QR code). Due to the limited capacity of these codes the transactions items themselves would not be included, but their hash value.

Due to technical limitations of certain printers (e.g. if a printer only prints uppercase or certain diacritics may not be printed) problems can arise with these requirements. To avoid this, a character substitution (→ 6.2) should be done before the hash value is computed.

10 Annex

10.1 Examples

Note: All examples in this section use the cash register profile.

10.1.1 Example: Transaction 1

The transaction items of the profile “cash register” are encoded according to the profile specifications. A SHA-1 hash value is then calculated according to the hash specifications for transaction items. (The transaction items are not transmitted to the TIM interface.) The use of the new cryptographic algorithms merely increases the number of characters for the hash value from 20 to 32.

Transaction items according to cash register profile (hex)*	**Contents**
A0h 04h 30h 2Eh 30h 38h	BP quantity : "0.08"
A1h 02h 6Bh 67h	BP unit : "kg"
A2h 0Bh 6Ah 61h 70h 61h 6Eh 73h 65h 6Eh 63h 68h 61h	BP name : "japansencha"
B2h 02h 47h 2Ch	BP price 2 : "+472"
A0h 01h 31h	BP quantity : "1"
A2h 10h 74h 65h 65h 6Bh 61h 6Eh 6Eh 65h 67h 75h 73h 73h 65h 69h 73h 65h	BP name : "teekannegusseise"
B1h 03h 04h 99h 0C	BP price 1 : "+4990"
A0h 01h 31h	BP quantity : "1"
A2h 09h 31h 30h 23h 72h 61h 62h 61h 74h 74h	BP name : "10#discount"
AAh 00h	BP flag discount/surcharge
B1h 02h 49h 9Dh	BP Price 1 : "-499"
5Eh F0h 13h F1h A1h F3h 3Bh 00h FBh 18h 00h 9Bh BCh 51h 63h 8Bh 36h 4Ch 6Eh 28h	SHA-1 hash value of the transaction items

* Paragraphs and indents serve only for illustration

After calculation of the hash value for the transaction items, the TRANSACTION command is composed and transmitted to the TIM. The TIM signs the data and returns the signature. The use of the new cryptographic algorithms merely increases the number of characters in the signature from 48 to 64.

Command / response (hex)*	Contents
80h 40h 00h 00h 50h CDh 04h 20h 10h 02h 28h CEh 02h 23h 59h C6h 09h 6Fh 70h 65h 72h 61h 74h 6Fh 72h 35h C7h 14h 5Eh F0h 13h F1h A1h F3h 3Bh 00h FBh 18h 00h 9Bh BCh 51h 63h 8Bh 36h 4Ch 6Eh 28h C8h 02h 03h D2h E1h 11h D8h 03h 04h 49h 1Ch D9h 02h 49h 9Ch DAh 02h 71h 7Ch DBh 02h 19h 00h E2h 0Ch D8h 02h 47h 2Ch DAh 02h 03h 1Ch DBh 02h 07h 00h 00h	Command TRANSACTION LC = 80 Byte Date : "2010-02-28" Time : "23-59" Operator : "operator5" Hash value of the transaction items Currency code: 978 (Euro) Container 1 Turnover "+4491" NegTurnover "+499" Value-added tax "+717" VAT rate "1900" Container 2 Turnover "+472" Value-added tax "+31" VAT rate "0700" LE = 00h
 C4h 0Fh 49h 4Eh 53h 49h 4Bh 41h 5Fh 54h 45h 53h 54h 5Fh 50h 54h 42h C5h 01h 01h CBh 02h 27h C0h 9Eh 30h 47h 40h 88h BAh D5h 4Dh B9h 48h 5Ch 93h 19h 29h F3h 0Bh 54h C7h 28h 9Eh C2h 6Ch F0h F1h 2Ah C2h 75h 70h 42h A4h 42h E0h 8Dh B1h A4h 0Ah 88h 27h 2Eh C8h 4Ch E4h 8Dh 33h B1h 32h 35h 75h 12h 19h 90h 00h	Response TPID:"INSIKA_TEST_PTB" TPIDNO: "1" Seq.No of the transaction: "10176" Signature SW1/SW2, No Error

* Paragraphs and indents serve only for illustration

10.1.2 Example: Transaction 2

Profile data:

Transaction items according to cash register profile (hex)*	Contents
A0h 05h 35h 34h 2Eh 30h 33h A1h 01h 6Ch A2h 06h 64h 69h 65h 73h 65h 6Ch ACh 00h B1h 03h 06h 21h 3C	ITEM QUANTITY: : "54.03" ITEM UNIT: "l" ITEM NAME: "diesel" ITEM THIRDPARTY: ITEM Preis 1: "+6213"
E0h 72h F8h C3h 1Dh 5Ch BDh 44h A8h C3h 7Bh 4Eh 80h DCh 63h 08h 16h 46h E2h 4Eh	SHA-1 hash value of the transaction items for SHA-256, 32 characters

* Paragraphs and indents serve only for illustration

Command TRANSACTION:

Command / response (hex) *	Contents
80h 40h 00h 00h 3Dh	Command TRANSACTION LC = 61 Byte
CDh 04h 20h 17h 02h 01h	Date: "2017-02-01"
CEh 02h 12h 00h	Time: "12:00"
C6h 03h 30h 30h 31h	Operator "001"
C7h 14h E0h 72h F8h C3h 1Dh 5Ch BDh 44h A8h C3h 7Bh 4Eh 80h DCh 63h 08h 16h 46h E2h 4Eh	Hash value of the transaction items For SHA-256, here 32 characters
C8h 02h 03h D2h	Currency code: 978 (Euro)
E1h 12h D8h 03h 06h 21h 3Ch DAh 02h 99h 2Ch DBh 02h 19h 00h D6h 03h 06h 21h 3Ch	Container 1 Turnover "+6213" Value-added tax "+992" VAT rate "1900" Container third-party Turnover "+6213"
00h	LE = 00h
	Response
C4h 10h 49h 4Eh 53h 49h 4Bh 41h 5Fh 54h 45h 53h 54h 5Fh 41h 44h 4Dh 5Ah	TPID:"INSIKA_TEST_ADMZ"
C5h 02h 03h E7h	TPIDNO: "999"
CBh 01h 02h	Seq.No of the transaction "2"
9Eh 30h 5Fh 10h D9h 4Fh 9Bh D0h A4h 97h 0Bh EEh ACh 25h 96h 40h 50h CBh 29h BBh 3Eh 7Eh D2h A0h A1h AAh 02h F8h FDh B3h 5Eh FBh 82h 57h F5h E4h 5Fh 5Dh 9Fh 1Ah ACh 7Fh 94h FCh 58h 47h 6Dh 88h EBh 6Bh	Signature 48 or 64 Byte, see footnote 1
90h 00h	SW1/SW2, No Error

Remark: Possible verification of data set above with the following publich key:

A87AD0A2D0B60C24F75DAE1CEFDABA64EBFEDDC603618683

B3C2B0944D317D14DCA2458DAA915577F1AEB62D8A886AD4

10.1.3 Example of Report

Signed report:

Command / response (hex) *	Contents
80h 42h 01h 00h 0Ah CDh 04h 20h 09h 05h 01h CEh 02h 17h 37h 00h	**Command REPORT signed** LC = 10 Byte Date: "2009-05-01" Time: "17-37" LE = 00h
 C0h 01h 03h C4h 0Dh 54h 50h 49h 44h 5Fh 54h 45h 53h 54h 5Fh 50h 54h 42h CCh 01h 1Ah C5h 01h 03h D2h 01h 01h D3h 01h 15h E1h 0Eh D8h 03h 02h 49h 9Ch D9h 03h 02h 52h 0Ch DBh 02h 19h 00h 9Eh 30h 4Bh A5h AEh E1h D4h E0h 10h ABh 37h 16h B4h 4Dh 78h 06h 2Bh 82h 14h 72h 3Fh 2Bh 4Bh 68h 06h 7Eh DCh F7h E5h 61h 69h 15h CAh 76h FBh 1Ch B5h 99h 71h 2Ah C9h C6h D7h F2h 97h 52h 46h 74h E9h 21h 90h 00h	**Response** Life cycle "TIM_ACTIVATED" TPID:"TPID_TEST_PTB" TPIDNO: "26" Seq.No of report: "3" Seq.No of first transaction: "1" Seq.No of last transaction: "176" Container 1 Turnover: " +2499" Negative turnover: "+2520" VAT rate: "1900" Signature 48 or 64 Byte, see footnote 1 SW1/SW2, No Error

* Paragraphs and indents serve only for illustration

10.1.4 Example: READ CERTIFICATE

In this example the certificate is read from the TIM. Blocks of 128 bytes (LE=80h) are requested for this. The offset (P1 and P2) of the READ CERTIFICATE command is incremented by 80h each time.

As long as the length of the certificate is not evaluated during this sequence of commands, the certificate can be read up to error message 6A 86h. The length of the certificate must then be evaluated in order to obtain the X.509 certificate.

Command / response (hex) *	Contents
00h A4h 00h 0Ch 02h 11h 10h	**Command: SELECT FILE** LC = 2 Byte Data = 11 10h (EF_CERT) LE = --
90h 00h	**Response** SW1/SW2: No Error
00h B0h 00h 00h 00h 80h	**Command** **READ CERTIFICATE** Offset = 00 00h LC = 0 Byte LE = 80h
30h 82h 03h 60h 30h 82h 02h 48h A0h 03h 02h 01h 02h 02h 04h 00h 00h 04h 50h 30h 0Dh 06h 09h 2Ah 86h 48h 86h F7h 0Dh 01h 01h 05h 05h 00h 30h 81h 84h 31h 0Bh 30h 09h 06h 03h 55h 04h 06h 13h 02h 44h 45h 31h 2Eh 30h 2Ch 06h 03h 55h 04h 0Ah 0Ch 25h 50h 68h 79h 73h 69h 6Bh 61h 6Ch 69h 73h 63h 68h 2Dh 54h 65h 63h 68h 6Eh 69h 73h 63h 68h 65h 20h 42h 75h 6Eh 64h 65h 73h 61h 6Eh 73h 74h 61h 6Ch 74h 31h 2Bh 30h 29h 06h 03h 55h 04h 0Bh 0Ch 22h 44h 61h 74h 65h 6Eh 6Bh 6Fh 6Dh 6Dh 75h 6Eh 69h 6Bh 61h 74h 69h 6Fh 6Eh 20h 90h 00h	**Response** Certificate (EF_CERT) 00 00h .. 00 7Fh Contents: The first bytes contain the length information 30h: SEQUENCE 82h: Length data in Byte 03h 60h: Length of certificate = 864 bytes SW1/SW2: No Error
00h B0h 00h 80h 00h 80h	**Command** **READ CERTIFICATE** Offset = 00 80h LC = 0 Byte LE = 80h
90h 00h	**Response** Certificate 00 80h .. 00 FFh SW1/SW2: No Error
00h B0h 01h 00h 00h 80h	**Command** **READ CERTIFICATE** Offset = 01 00h LC = 0 Byte LE = 80h
..	
SW1/SW2=6Ah 86h (Checking error: Incorrect P1-P2) Lr=0	

* Paragraphs and indents serve only for illustration

10.2 Sequence Diagrams

10.2.1 Example: First initialisation of the TIM

The following example shows the procedure for the first initialisation. First the TIM application is selected with SELECT FILE (AID). The following GET DATA TIM Status checks the life cycle, in this example TIM_PERSONALISED. The TIM must therefore be activated once before the first transaction. The transport PIN is requested from the user. The transport PIN is transmitted to the TIM with the command REPORT TIM Activate. If the check is successful, the TIM sends back a signed report. Information such as TIM version, tax payer ID, sequence number of the TIM, etc. is then polled with the command GET DATA TIM Status extended and stored in the journal of the cash register. The certificate on the TIM is selected with SELECT FILE (EF_CERT) and then read out with the commands READ CERTIFICATE part 1..N. The certificate is also stored in the journal of the cash register and later inserted into the XML export file(s).

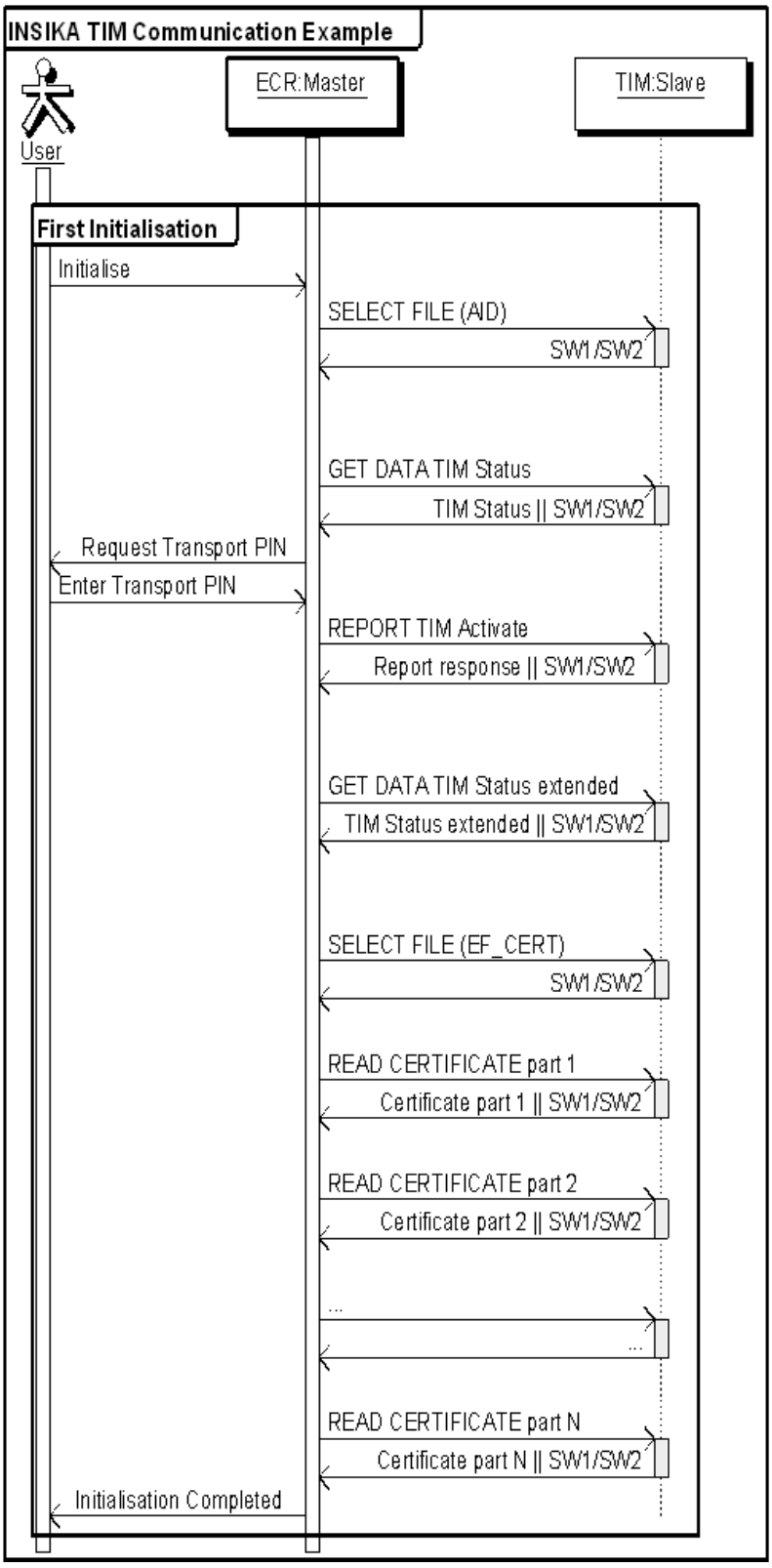

Figure 10-1: Sequence diagram for first initialisation of the TIM (example)

Note: The SELECT FILE (AID) command is no longer required from TIM version V.2.0.0..

10.2.2 Example: Transactions and report

An example of the procedure during normal use is shown below. The TIM application is first selected with SELECT FILE. Transactions are then signed with the command TRANSACTION and a report is received with the command REPORT signed.

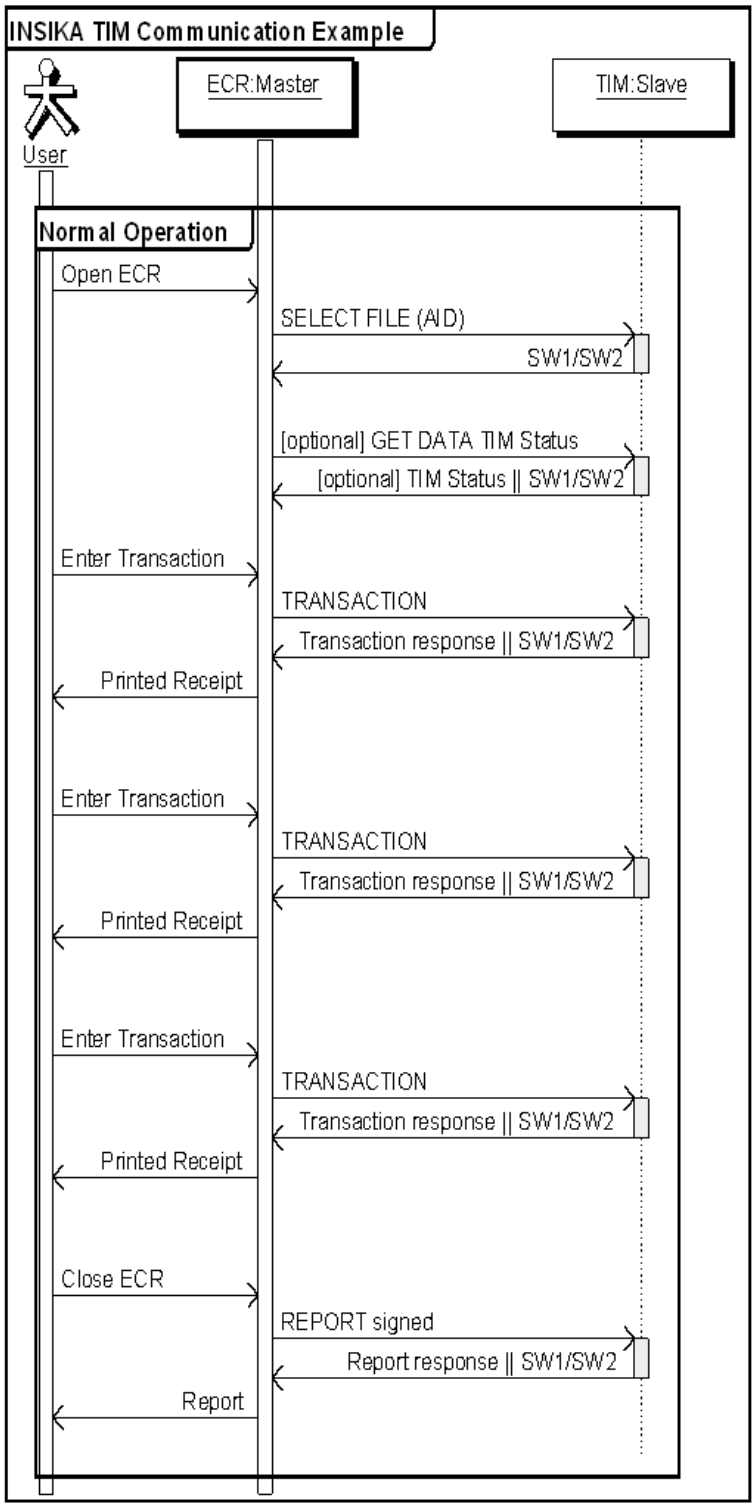

Figure 10-2: Sequence diagram with normal use of the TIM (example)

Note: The SELECT FILE (AID) command is no longer required from TIM version V.2.0.0..

10.2.3 Example: Deactivation of the TIM

The TIM can only be deactivated once. The command TRANSACTION is then no longer available (see → 5.3). Renewed activation of the TIM is not possible!

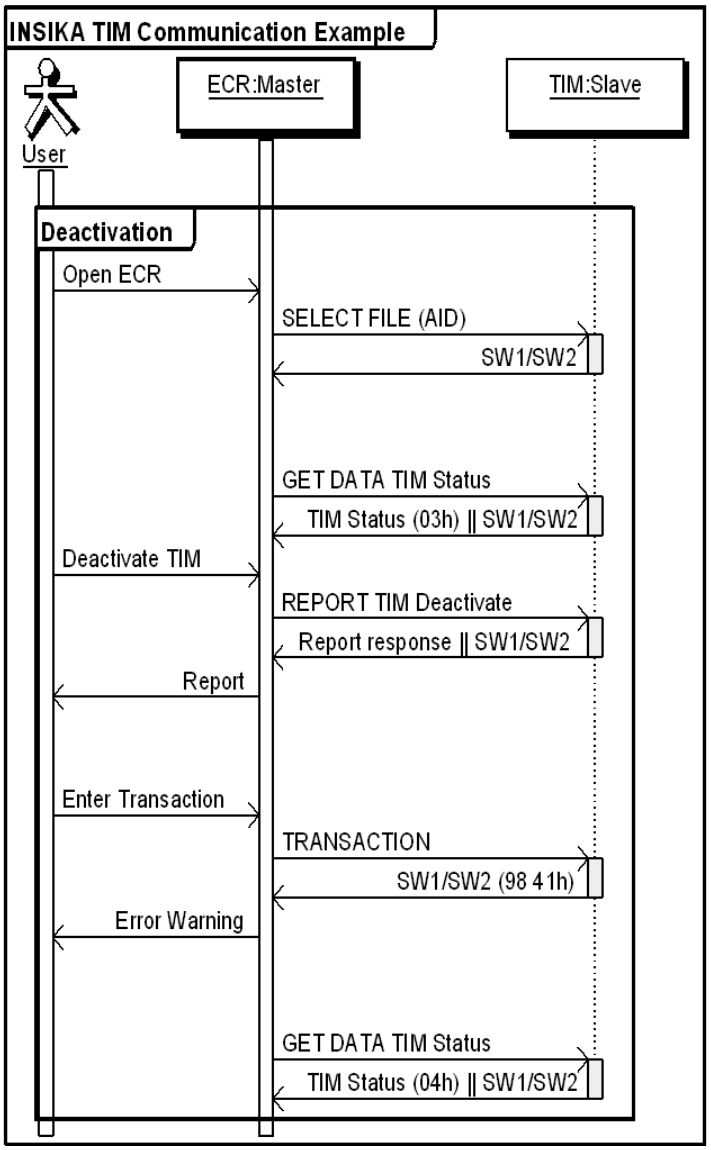

Figure 10-3: Sequence diagram for deactivation of the TIM (example)

Note: The SELECT FILE (AID) command is no longer required from TIM version V.2.0.0..