SASO IEC 62055-41:2019
IEC 62055-41:2018

ELECTRICITY METERING – PAYMENT SYSTEMS – Part 41: Standard transfer specification (STS) – Application layer protocol for one-way token carrier systems

ICS 17.220.20

Saudi Standards, Metrology and Quality Org (SASO)

---

this document is a draft saudi standard circulated for comment. it is, therefore subject to change and may not be referred to as a saudi standard until approved by the board of directors.

DRAFT

Foreword

The Saudi Standards ,Metrology and Quality Organization (SASO)has adopted the International standard No. IEC 62055-41:2018 “ELECTRICITY METERING – PAYMENT SYSTEMS – Part 41: Standard transfer specification (STS) – Application layer protocol for one-way token carrier systems” issued by (IEC). The text of this international standard has been translated into Arabic so as to be approved as a Saudi standard.

DRAFT

DRAFT

CONTENTS

FOREWORD 8
INTRODUCTION 10
1 Scope 13
2 Normative references 13
3 Terms, definitions, abbreviated terms, notation and terminology 14
3.1 Terms and definitions 14
3.2 Abbreviated terms 16
3.3 Notation and terminology 18
4 Numbering conventions 18
5 Reference model for the standard transfer specification 19
5.1 Generic payment meter functional reference diagram 19
5.2 STS protocol reference model 20
5.3 Dataflow from the POSApplicationProcess to the TokenCarrier 21
5.4 Dataflow from the TokenCarrier to the MeterApplicationProcess 21
5.5 MeterFunctionObjects / companion specifications 23
5.6 Transaction reference numbers 23
6 POSToTokenCarrierInterface application layer protocol 23
6.1 APDU: ApplicationProtocolDataUnit 23
6.1.1 Data elements in the APDU 23
6.1.2 MeterPAN: MeterPrimaryAccountNumber 25
6.1.3 TCT: TokenCarrierType 26
6.1.4 DKGA: DecoderKeyGenerationAlgorithm 27
6.1.5 EA: EncryptionAlgorithm 27
6.1.6 SGC: SupplyGroupCode 27
6.1.7 TI: TariffIndex 28
6.1.8 KRN: KeyRevisionNumber 28
6.1.9 KT: KeyType 28
6.1.10 KEN: KeyExpiryNumber 29
6.1.11 DOE: DateOfExpiry 29
6.1.12 BDT: BaseDate 29
6.2 Tokens 30
6.2.1 Token definition format 30
6.2.2 Class 0: TransferCredit 30
6.2.3 Class 1: InitiateMeterTest/Display 31
6.2.4 Class 2: SetMaximumPowerLimit 31
6.2.5 Class 2: ClearCredit 31
6.2.6 Class 2: SetTariffRate 31
6.2.7 Key change token set for 64-bit DecoderKey transfer 32
6.2.8 Key change token set for 128-bit DecoderKey transfer 33
6.2.9 Class 2: ClearTamperCondition 34
6.2.10 Class 2: SetMaximumPhasePowerUnbalanceLimit 34
6.2.11 Class 2: SetWaterMeterFactor 34
6.2.12 Class 2: Reserved for STS use 34
6.2.13 Class 2: Reserved for Proprietary use 35

DRAFT

6.2.14 Class 3: Reserved for STS use 35
6.3 Token data elements 35
6.3.1 Data elements used in tokens 35
6.3.2 Class: TokenClass 36
6.3.3 SubClass: TokenSubClass 37
6.3.4 RND: RandomNumber 37
6.3.5 TID: TokenIdentifier 38
6.3.6 Amount: TransferAmount 39
6.3.7 CRC: CyclicRedundancyCheck 43
6.3.8 Control: InitiateMeterTest/DisplayControlField 44
6.3.9 MPL: MaximumPowerLimit 45
6.3.10 MPPUL: MaximumPhasePowerUnbalanceLimit 45
6.3.11 Rate: TariffRate 45
6.3.12 WMFactor: WaterMeterFactor 45
6.3.13 Register: RegisterToClear 45
6.3.14 NKHO: NewKeyHighOrder 45
6.3.15 NKLO: NewKeyLowOrder 45
6.3.16 NKMO1: NewKeyMiddleOrder1 45
6.3.17 NKMO2: NewKeyMiddleOrder2 46
6.3.18 KENHO: KeyExpiryNumberHighOrder 46
6.3.19 KENLO: KeyExpiryNumberLowOrder 46
6.3.20 RO: RolloverKeyChange 46
6.3.21 S&E: SignAndExponent 46
6.3.22 CRC_C: CyclicRedundancyCheck_C 46
6.4 TCDUGeneration functions 46
6.4.1 Definition of the TCDU 46
6.4.2 Transposition of the Class bits 47
6.4.3 TCDUGeneration function for Class 0,1 and 2 tokens 47
6.4.4 TCDUGeneration function for key change tokens 49
6.4.5 TCDUGeneration function for Set2ndSectionDecoderKey token 50
6.5 Security functions 50
6.5.1 General requirements 50
6.5.2 Key attributes and key changes 50
6.5.3 DecoderKey generation 58
6.5.4 STA: EncryptionAlgorithm07 65
6.5.5 DEA: EncryptionAlgorithm09 68
6.5.6 MISTY1: EncryptionAlgorithm11 68
7 TokenCarriertoMeterInterface application layer protocol 70
7.1 APDU: ApplicationProtocolDataUnit 70
7.1.1 Data elements in the APDU 70
7.1.2 Token 71
7.1.3 AuthenticationResult 71
7.1.4 ValidationResult 71
7.1.5 TokenResult 72
7.2 APDUExtraction functions 73
7.2.1 Extraction process 73
7.2.2 Extraction of the 2 Class bits 73
7.2.3 APDUExtraction function for Class 0 and Class 2 tokens 74

7.2.4 APDUExtraction function for Class 1 tokens 75
7.2.5 APDUExtraction function for key change token set 75
7.3 Security functions 76
7.3.1 Key attributes and key changes 76
7.3.2 DKR: DecoderKeyRegister 76
7.3.3 STA: DecryptionAlgorithm07 77
7.3.4 DEA: DecryptionAlgorithm09 80
7.3.5 MISTY1: DecryptionAlgorithm11 80
7.3.6 TokenAuthentication 82
7.3.7 TokenValidation 82
7.3.8 TokenCancellation 83
8 MeterApplicationProcess requirements 83
8.1 General requirements 83
8.2 Token acceptance/rejection 84
8.3 Display indicators and markings 85
8.4 TransferCredit tokens 85
8.5 InitiateMeterTest/Display tokens 85
8.6 SetMaximumPowerLimit tokens 86
8.7 ClearCredit tokens 86
8.8 SetTariffRate tokens 86
8.9 Key change tokens 86
8.10 Set2ndSectionDecoderKey tokens 87
8.11 ClearTamperCondition tokens 87
8.12 SetMaximumPhasePowerUnbalanceLimit tokens 87
8.13 SetWaterMeterFactor 87
8.14 Class 2: Reserved for STS use tokens 87
8.15 Class 2: Reserved for Proprietary use tokens 87
8.16 Class 3: Reserved for STS use tokens 88
9 KMS: KeyManagementSystem generic requirements 88
10 Maintenance of STS entities and related services 88
10.1 General 88
10.2 Operations 90
10.2.1 Product certification maintenance 90
10.2.2 DSN maintenance 90
10.2.3 RO maintenance 90
10.2.4 TI maintenance 90
10.2.5 TID maintenance 91
10.2.6 SpecialReservedTokenIdentifier maintenance 91
10.2.7 MfrCode maintenance 91
10.2.8 Substitution tables maintenance 91
10.2.9 Permutation tables maintenance 91
10.2.10 SGC maintenance 91
10.2.11 VendingKey maintenance 91
10.2.12 KRN maintenance 91
10.2.13 KT maintenance 91
10.2.14 KEN maintenance 92
10.2.15 CERT maintenance 92
10.2.16 CC maintenance 92

DRAFT

10.2.17 UC maintenance ... 92
10.2.18 KMCID maintenance ... 92
10.2.19 CMID maintenance ... 92
10.3 Standardisation ... 92
10.3.1 IIN maintenance ... 92
10.3.2 TCT maintenance ... 93
10.3.3 DKGA maintenance ... 93
10.3.4 EA maintenance ... 93
10.3.5 TokenClass maintenance ... 93
10.3.6 TokenSubClass maintenance ... 93
10.3.7 InitiateMeterTest/DisplayControlField maintenance ... 93
10.3.8 RegisterToClear maintenance ... 94
10.3.9 STS BaseDate maintenance ... 94
10.3.10 Rate maintenance ... 94
10.3.11 WMFactor maintenance ... 94
10.3.12 MFO maintenance ... 94
10.3.13 FOIN maintenance ... 95
10.3.14 Companion specification maintenance ... 95
Annex A (informative) Guidelines for a KeyManagementSystem (KMS) ... 96
Annex B (informative) Entities and identifiers in an STS-compliant system ... 100
Annex C (informative) Code of practice for the implementation of STS-compliant systems ... 104
C.1 General ... 104
C.2 Maintenance and support services provided by the STS Association ... 104
C.3 Key management ... 104
C.3.1 Key management services ... 104
C.3.2 SupplyGroupCode and VendingKey distribution ... 104
C.3.3 CryptographicModule distribution ... 105
C.3.4 Key expiry ... 106
C.4 MeterPAN ... 106
C.4.1 General practice ... 106
C.4.2 IssuerIdentificationNumbers ... 106
C.4.3 ManufacturerCodes ... 106
C.4.4 DecoderSerialNumbers ... 107
C.5 SpecialReservedTokenIdentifier ... 107
C.6 Permutation and substitution tables for the STA ... 107
C.7 EA codes ... 107
C.8 TokenCarrierType codes ... 107
C.9 MeterFunctionObject instances / companion specifications ... 108
C.10 TariffIndex ... 108
C.11 STS-compliance certification ... 108
C.11.1 IEC certification services ... 108
C.11.2 Products ... 108
C.11.3 Certification authority ... 108
C.12 Procurement options for users of STS-compliant systems ... 108
C.13 Management of TID roll over ... 112
C.13.1 Introduction ... 112
C.13.2 Overview ... 113

DRAFT

C.13.3 Impact analysis 114
C.13.4 Base dates 115
C.13.5 Implementation 115
Bibliography 118

Figure 1 – Functional block diagram of a generic single-device payment meter 19
Figure 2 – STS modelled as a 2-layer collapsed OSI protocol stack 20
Figure 3 – Dataflow from the POSApplicationProcess to the TokenCarrier 21
Figure 4 – Dataflow from the TokenCarrier to the MeterApplicationProcess 22
Figure 5 – Composition of transaction reference number 23
Figure 6 – Transposition of the 2 Class bits 47
Figure 7 – TCDUGeneration function for Class 0, 1 and 2 tokens 48
Figure 8 – TCDUGeneration function for key change tokens 49
Figure 9 – DecoderKey changes – state diagram 56
Figure 10 – DecoderKeyGenerationAlgorithm01 61
Figure 11 – DecoderKeyGenerationAlgorithm02 62
Figure 12 – STA: EncryptionAlgorithm07 65
Figure 13 – STA encryption substitution process 66
Figure 14 – STA encryption permutation process 67
Figure 15 – STA encryption DecoderKey rotation process 67
Figure 16 – STA encryption worked example for TransferCredit token 68
Figure 17 – MISTY1: EncryptionAlgorithm11 69
Figure 18 – MISTY1 encryption worked example for TransferCredit token 70
Figure 19 – APDUExtraction function 73
Figure 20 – Extraction of the 2 Class bits 74
Figure 21 – STA DecryptionAlgorithm07 77
Figure 22 – STA decryption permutation process 77
Figure 23 – STA decryption substitution process 78
Figure 24 – STA decryption DecoderKey rotation process 79
Figure 25 – STA decryption worked example for TransferCredit token 80
Figure 26 – STA DecryptionAlgorithm11 81
Figure 27 – MISTY1 decryption worked example for TransferCredit token 81
Figure A.1 – KeyManagementSystem and interactive relationships between entities 96
Figure B.1 – Entities and identifiers deployed in an STS-compliant system 100
Figure C.1 – System overview 113

Table 1 – Data elements in the APDU 24
Table 2 – Data elements in the IDRecord 24
Table 3 – Data elements in the MeterPAN 25
Table 4 – Data elements in the IAIN / DRN 25
Table 5 – Token carrier types 26
Table 6 – DKGA codes 27
Table 7 – EA codes 27

DRAFT

Table 8 – SGC types and key types 28
Table 9 – DOE codes for the year 29
Table 10 – DOE codes for the month 29
Table 11 – BDT representation 30
Table 12 – Token definition format 30
Table 13 – Data elements used in tokens 35
Table 14 – Token classes 36
Table 15 – Token sub-classes 37
Table 16 – TID calculation examples 38
Table 17 – Units of measure for electricity 39
Table 18 – Units of measure for other applications 40
Table 19 – Bit allocations for the Amount field for SubClass 0 to 3 40
Table 20 – Maximum error due to rounding 41
Table 21 – Examples of TransferAmount values for credit transfer 41
Table 22 – Bit allocations for the Amount field for SubClass 4 to 7 41
Table 23 – Bit allocations for the exponent e 41
Table 24 – Examples of rounding of negative and positive values 42
Table 25 – Examples of TransferAmounts and rounding errors 43
Table 26 – Example of a CRC calculation 43
Table 27 – Permissible control field values 44
Table 28 – Selection of register to clear 45
Table 29 – S&E bit positions for variables s, e_4, e_3 and e_2 46
Table 30 – Example of a CRC_C calculation 46
Table 31 – Classification of vending keys 52
Table 32 – Classification of decoder keys 52
Table 33 – Permitted relationships between decoder key types 57
Table 34 – Definition of the PANBlock 59
Table 35 – Data elements in the PANBlock 59
Table 36 – Definition of the CONTROLBlock 59
Table 37 – Data elements in the CONTROLBlock 59
Table 38 – Range of applicable decoder reference numbers 60
Table 39 – List of applicable supply group codes 61
Table 40 – Data elements in DataBlock 63
Table 41 – Input parameters for a worked example 64
Table 42 – DataBlock example construction 64
Table 43 – DecoderKey construction example 64
Table 44 – Sample substitution tables 66
Table 45 – Sample permutation table 67
Table 46 – Data elements in the APDU 71
Table 47 – Possible values for the AuthenticationResult 71
Table 48 – Possible values for the ValidationResult 72
Table 49 – Possible values for the TokenResult 72
Table 50 – Values stored in the DKR 76

DRAFT

Table 51 – Sample permutation table 78
Table 52 – Sample substitution tables 79
Table 53 – Entities/services requiring maintenance service 89
Table A.1 – Entities that participate in KMS processes 97
Table A.2 – Processes surrounding the payment meter and DecoderKey 97
Table A.3 – Processes surrounding the CryptographicModule 98
Table A.4 – Processes surrounding the SGC and VendingKey 98
Table B.1 – Typical entities deployed in an STS-compliant system 101
Table B.2 – Identifiers associated with the entities in an STS-compliant system 102
Table C.1 – Data elements associated with a SGC 105
Table C.2 – Data elements associated with the CryptographicModule 106
Table C.3 – Items that should be noted in purchase orders and tenders 109

DRAFT

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

ELECTRICITY METERING – PAYMENT SYSTEMS –

Part 41: Standard transfer specification (STS) – Application layer protocol for one-way token carrier systems

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62055-41 has been prepared by IEC technical committee 13: Electrical energy measurement and control.

This third edition cancels and replaces the second edition of IEC 62055-41, issued in 2014. It constitutes a technical revision.

The main technical changes with regard to the previous edition are as follows:

- currency transfer tokens for electricity, water, gas and time metering;
- finer resolution for gas and time credit transfer;
- common code PAN for 2 and 4 digit manufacturer codes;
- reserved MfrCode values for certification and testing purposes;
- provision for DLMS/COSEM as a virtual token carrier type;
- addition of DKGA04, an advanced key derivation function from 160-bit VendingKey;

- withdrawal of DES for EA09 and TDES for DKGA03 cryptographic algorithms, but DES for DKGA02 remains in use;
- addition of MISTY1 cryptographic algorithm using a 128-bit DecoderKey with supporting key change tokens;
- transfer of SGC values to the meter via key change tokens;
- revision of the test/display token requirements;
- revision of the KMS to reflect current best practice;
- revision of the TID roll over management guidelines;
- definition of BaseDate is referenced to Coordinated Universal Time;
- disassociation of IIN from the ISO standard definition;
- various clarifications and enhancements to support the above.

The text of this standard is based on the following documents:

FDIS	Report on voting
13/1755/FDIS	13/1764/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62055 series, published under the general title *Electricity metering – Payment systems*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under "http://webstore.iec.ch" in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

DRAFT

INTRODUCTION

The IEC 62055 series covers payment systems, encompassing the customer information systems, point of sale systems, token carriers, payment meters and the respective interfaces that exist between these entities. At the time of preparation of this document, IEC 62055 comprised the following parts, under the general title, *Electricity metering – Payment systems:*

Part 21: Framework for standardization

Part 31: Particular requirements – Static payment meters for active energy (classes 1 and 2)

Part 41: Standard transfer specification (STS) – Application layer protocol for one-way token carrier systems

Part 42: Transfer reference numbers (TRN) – Application layer protocol for one-way token carrier systems

Part 51: Standard transfer specification (STS) – Physical layer protocol for one-way numeric and magnetic card token carriers

Part 52: Standard transfer specification (STS) – Physical layer protocol for a two-way virtual token carrier for direct local connection

Part 4x series specify application layer protocols and Part 5x series specify physical layer protocols.

NOTE 1 Part 42 is not interoperable with Part 41, Part 51 and Part 52.

NOTE 2 Part 42 was in preparation at the time of publication of this edition of Part 41.

The standard transfer specification (STS) is a secure message protocol that allows information to be carried between point of sale (POS) equipment and payment meters and it caters for several message types such as credit, configuration control, display and test instructions. It further specifies devices and codes of practice that allow for the secure management (generation, storage, retrieval and transportation) of cryptographic keys used within the system.

The token carrier, which is not specified in this part of IEC 62055, is the physical device or medium used to transport the information from the POS equipment to the payment meter. Three types of token carriers are currently specified in IEC 62055-51 and IEC 62055-52; the magnetic card, the numeric token carrier and a virtual token carrier, which have been approved by the STS Association. New token carriers can be proposed as new work items through the National Committees or through the STS Association.

Although the main implementation of the STS is in the electricity supply industry, it inherently provides for the management of other utility services such as water and gas. It should be noted that certain functionalities may not apply across all utility services, for example, MaximumPowerLimit in the case of a water meter. Similarly, certain terminology may not be appropriate in non-electrical applications, for example, Load Switch in the case of a gas meter. Future revisions of the STS may allow for other token carrier technologies like smart cards and memory keys with two-way functionality and to cater for a real-time clock and complex tariffs in the payment meter.

Not all the requirements specified in this document are compulsory for implementation in a particular system configuration and as a guideline, a selection of optional configuration parameters are listed in Clause C.12.

The STS Association is registered with the IEC as a Registration Authority for providing maintenance services in support of the STS (see Clause C.1 for more information).

Publication of the first edition of IEC 62055-41 in May 2007 resulted in its rapid adoption as the preferred global standard for prepayment meters in many IEC member countries and a majority of IEC affiliate member countries. Prepayment electricity meters and their associated Payment Systems are now produced, operated and maintained by an ecosystem of utilities, meter

DRAFT

manufacturers, meter operators, vending system providers, vending agents, banking institutions and adjacent industries. Multi-stakeholder interests are served by the STS Association comprising of more than 150 organisations located in over 35 countries. Interoperability and conformance to the Standard Transfer Specification (STS) are guaranteed by Conformance test specifications developed and administered by the STS Association. A full list of the STS Association services can be found at http://www.sts.org.za.

Developed originally for prepayment electricity meters in Africa – via an IEC TC13 WG15 D-type liaison with the STS Association – this IEC standard now serves more users in Asia than Africa, with a total of approximately 50 million meters operated by 500 utilities in 94 countries. Management of the technology has been administered by the STS Association in fulfilment of its role as the IEC appointed Registration Authority.

With the ongoing development of advanced cryptographic algorithms, it has become desirable to revise the security levels of IEC 62055-41 so as to reflect the state of the art best practices, which will be appropriate for deployment of new systems having a useful life expectancy of at least the next 30 years.

Similarly, smart metering systems with payment functionality have evolved to employ tariff functions in the meter, thus raising the need to provide for the transfer of currency units to the meter instead of service units.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent concerning special reserved token identifier given in 6.3.5.2.

IEC takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured the IEC that he/she is willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with IEC. Information may be obtained from:

Address:	Itron Measurement and Systems, P.O. Box 4059,TygerValley 7536, Republic of South Africa
Tel:	+27 21 928 1700
Fax:	+27 21 928 1701
Website:	http://www.itron.com

Address:	Conlog (Pty) Ltd, P.O. Box 2332, Durban 4000, Republic of South Africa
Tel:	+27 31 2681141
Fax:	+27 31 2087790
Website:	http://www.conlog.co.za

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (http://patents.iec.ch) maintain on-line data bases of patents relevant to their standards. Users are encouraged to consult the data bases for the most up to date information concerning patents.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this International Standard may involve the use of a maintenance service concerning encryption key management and the stack of protocols on which the present

DRAFT

International Standard IEC 62055-41 is based [see Clause C.1]. The IEC takes no position concerning the evidence, validity and scope of this maintenance service.

The provider of the maintenance service has assured the IEC that he is willing to provide services under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the provider of the maintenance service is registered with the IEC. Information may be obtained from:

Address:	The STS Association, P.O. Box 868, Ferndale 2160, Republic of South Africa
Tel:	+27 11 061 5000
Fax:	+27 86 679 4500
Email:	support@sts.org.za
Website:	http://www.sts.org.za

DRAFT

ELECTRICITY METERING – PAYMENT SYSTEMS –

Part 41: Standard transfer specification (STS) – Application layer protocol for one-way token carrier systems

1 Scope

This part of IEC 62055 specifies the application layer protocol of the STS for transferring units of credit and other management information from a point of sale (POS) system to an STS-compliant payment meter in a one-way token carrier system. It is primarily intended for application with electricity payment meters without a tariff employing energy-based tokens, but may also have application with currency-based token systems and for services other than electricity.

It specifies:

- a POS to token carrier interface structured with an application layer protocol and a physical layer protocol using the OSI model as reference;
- tokens for the application layer protocol to transfer the various messages from the POS to the payment meter;
- security functions and processes in the application layer protocol such as the Standard Transfer Algorithm and the Data Encryption Algorithm, including the generation and distribution of the associated cryptographic keys;
- security functions and processes in the application layer protocol at the payment meter such as decryption algorithms, token authentication, validation and cancellation;
- specific requirements for the meter application process in response to tokens received;
- a scheme for dealing with payment meter functionality in the meter application process and associated companion specifications;
- generic requirements for an STS-compliant key management system;
- guidelines for a key management system;
- entities and identifiers used in an STS system;
- code of practice for the management of TID roll-over key changes in association with the revised set of base dates;
- code of practice and maintenance support services from the STS Association.

It is intended for use by manufacturers of payment meters that have to accept tokens that comply with the STS and also by manufacturers of POS systems that have to produce STS-compliant tokens and is to be read in conjunction with IEC 62055-5x series.

STS-compliant products are required to comply with selective parts of this document only, which is the subject of the purchase contract (see also Clause C.12).

NOTE Although developed for payment systems for electricity, the document also makes provision for tokens used in other utility services, such as water and gas.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies.

DRAFT

For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC TR 62051:1999, *Electricity metering – Glossary of terms*

IEC TR 62055-21:2005, *Electricity metering – Payment systems – Part 21: Framework for standardization*

IEC 62055-31:2005, *Electricity metering – Payment systems – Part 31: Particular requirements – Static payment meters for active energy (classes 1 and 2)*

IEC 62055-51:2007, *Electricity metering – Payment systems – Part 51: Standard transfer specification (STS) – Physical layer protocol for one-way numeric and magnetic card token carriers*

IEC 62055-52:2008, *Electricity metering – Payment systems – Part 52: Standard transfer specification (STS) – Physical layer protocol for a two-way virtual token carrier for direct local connection*

ISO/IEC 7812-1:2017, *Identification cards – Identification of issuers – Part 1: Numbering system*

ISO/IEC 18033-3, *Information technology – Security techniques – Encryption Algorithms – Part 3: Block ciphers*

ISO 9797-2, *Information technology – Security techniques – Message Authentication. Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function*

ISO 10118-3, *Information technology – Security techniques – Hash-functions – Part 3: Dedicated Hash Functions*

ANSI X3.92-1981, *American National Standard Data Encryption Algorithm, American National Standards Institute – Data Encryption Algorithm*

FIPS PUB 46-3:1999, *Federal Information Processing Standards Publication – Data Encryption Standard*

NIST SP 800-108, *Recommendation for Key Derivation Using Pseudorandom Functions*

3 Terms, definitions, abbreviated terms, notation and terminology

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC TR 62051 and IEC 62055-31 as well as the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

NOTE Where there is a difference between the definitions in this document and those contained in other referenced IEC standards, then those defined in this document take precedence.

The term "meter" is used interchangeably with "payment meter", "prepayment meter" and "decoder", where the decoder is a sub-part of an electricity payment meter or of a multi-device payment meter.

DRAFT

The term “POS” is used synonymously with “CIS”, “MIS” and “HHU” in the sense that tokens may also be generated by, and transferred between these entities and the payment meter.

The term “utility” is used to signify the supplier of the service in a general sense. In the liberalized markets the actual contracting party acting as the “supplier” of the service to the consumer may not be the traditional utility as such, but may be a third party service provider.

3.1.1
companion specification
specification managed by the STS Association, which defines a specific instance of a MeterFunctionObject

SEE: 5.5 and Clause C.9.

3.1.2
decoder
part of the TokenCarrierToMeterInterface of a payment meter that performs the functions of the application layer protocol and which allows token-based transactions to take place between a POS and the payment meter

3.1.3
meter serial number
number that is associated with the metrological part of the payment meter

Note 1 to entry: In a single-device payment meter the DRN and meter serial number may be synonymous, while in a multi-device payment meter they may be different.

3.1.4
token
subset of data elements, containing an instruction and information that is present in the APDU of the Application Layer of the POSToTokenCarrierInterface, and which is also transferred to the payment meter by means of a token carrier (the converse is also true in the case of a token being sent from the payment meter to the POS)

3.1.5
token carrier
medium that is used in the Physical Layer of the POSToTokenCarrierInterface, onto which a token is modulated or encoded, and which serves to carry a token from the point where it is generated to the remote payment meter, where it is received

3.1.6
one-way token carrier system
payment metering system, which employs token carriers that transfer information in one direction only – from the POS to the payment meter

3.1.7
token-based transaction
processing of any token by the payment meter that has material effect on the amount, value or quality of service to be delivered to the consumer under control of the payment meter (in terms of current practice this means tokens of Class 0 and Class 2)

3.1.8
supported
ability to perform a defined function

Note 1 to entry: If a supported function is disabled, it remains supported.

3.1.9
base currency
particular currency denomination for the country that the receiving meter account is operating in, as defined in ISO 4217

DRAFT

EXAMPLES USD/840, EUR/978, GBP/826, ZAR/710.

3.2 Abbreviated terms

ANSI	American National Standards Institute
APDU	ApplicationProtocolDataUnit
BDT	BaseDate
CA	CertificationAuthority
CC	CountryCode
CERT	Certified public key
CIS	Customer Information System
CM	CryptographicModule
CMID	CryptographicModuleIdentifier
COP	Code of practice
COSEM	Companion Specification for Energy Metering
CRC	CyclicRedundancyCheck
DAC	DeviceAuthenticationCode
DCTK	DecoderCommonTransferKey
DD	Discretionary Data
DDTK	DecoderDefaultTransferKey
DEA	Data Encryption Algorithm
DES	Data Encryption Standard
DITK	DecoderInitializationTransferKey
DK	DecoderKey
DKGA	DecoderKeyGenerationAlgorithm
DKR	DecoderKeyRegister
DLMS	Distribution Line Message Specification
DOE	DateOfExpiry
DRN	DecoderReferenceNumber [known as a “meter number” in systems in use prior to the development of this document]
DSN	DecoderSerialNumber
DUTK	DecoderUniqueTransferKey
EA	EncryptionAlgorithm
ECB	Electronic Code Book
ETX	ASCII End of Text character
FAC	FirmwareAuthenticationCode
FIPS	Federal Information Processing Standards
FOIN	FunctionObjectIdentificationNumber
FS	FieldSeparator
GPRS	General Packet Radio Service
GSM	Global System For Mobile Communications
HHU	HandHeldUnit
HMAC	Hash Message Authentication Code
IAIN	IndividualAccountIdentificationNumber
ID	Identification; Identifier

DRAFT

IIN	IssuerIdentificationNumber
ISDN	Integrated Services Digital Network
ISO	International Standards Organisation
KCT	KeyChangeToken
KDF	Key Derivation Function
KEK	KeyExchangeKey
KEN	KeyExpiryNumber
KLF	KeyLoadFile
KMC	KeyManagementCentre
KMI	KeyManagementInfrastructure
KMS	KeyManagementSystem
KRN	KeyRevisionNumber
KT	KeyType
LAN	Local Area Network
LRC	LongitudinalRedundancyCheck
MFO	MeterFunctionObject
Mfr	Manufacturer
MII	MajorIndustryIdentifier
MIS	Management Information System
MPL	MaximumPowerLimit
MPPUL	MaximumPhasePowerUnbalanceLimit
NIST	National Institute of Standards and Technology
NKHO	NewKeyHighOrder bits
NKLO	NewKeyLowOrder bits
NWIP	New Work Item Proposal
OSI	Open Systems Interconnection
PAN	PrimaryAccountNumber
PLC	Power Line Carrier
POS	PointOfSale
PRN	Printer
PSTN	Public Switched Telephone Network
RND	RandomNumber
RO	Roll over
SG	SupplyGroup
SGC	SupplyGroupCode
SHA	Secure Hash Algorithm
STA	Standard Transfer Algorithm
STS	Standard Transfer Specification
STSA	Standard Transfer Specification Association
STX	ASCII Start of Text character
TCDU	TokenCarrierDataUnit
TCT	TokenCarrierType
TDEA	Triple Data Encryption Algorithm

DRAFT

TI	TariffIndex
TID	TokenIdentifier
UC	UtilityCode
VCDK	VendingCommonDerivationKey
VDDK	VendingDefaultDerivationKey
VK	VendingKey
VUDK	VendingUniqueDerivationKey
WAN	Wide Area Network
XOR	Exclusive Or (logical)

3.3 Notation and terminology

Throughout this document the following rules are observed regarding the naming of terms:

- entity names, data element names, function names and process names are treated as generic object classes and are given names in terms of phrases in which the words are capitalized and joined without spaces. Examples are: SupplyGroupCode as a data element name, EncryptionAlgorithm07 as a function name and TransferCredit as a process name (see note);
- direct (specific) reference to a named class of object uses the capitalized form, while general (non-specific) reference uses the conventional text i.e. lower case form with spaces. An example of a direct reference is: “The SupplyGroupCode is linked to a group of meters”, while an example of a general reference is: “A supply group code links to a vending key”;
- other terms use the generally accepted abbreviated forms like PSTN for Public Switched Telephone Network.

NOTE The notation used for naming of objects has been aligned with the so called “camel-notation” used in the common information model (CIM) standards prepared by IEC TC 57, in order to facilitate future harmonization and integration of payment system standards with the CIM standards.

4 Numbering conventions

In this document, the representation of numbers in binary strings uses the convention that the least significant bit is to the right, and the most significant bit is to the left.

Numbering of bit positions start with bit position 0, which corresponds to the least significant bit of a binary number.

Numbers are generally in decimal format, unless otherwise indicated. Any digit without an indicator signifies decimal format.

Binary digit values range from 0 to 1.

Decimal digit values range from 0 to 9.

Hexadecimal digit values range from 0 to 9, A to F and are indicated by “hex”.

DRAFT

5 Reference model for the standard transfer specification

5.1 Generic payment meter functional reference diagram

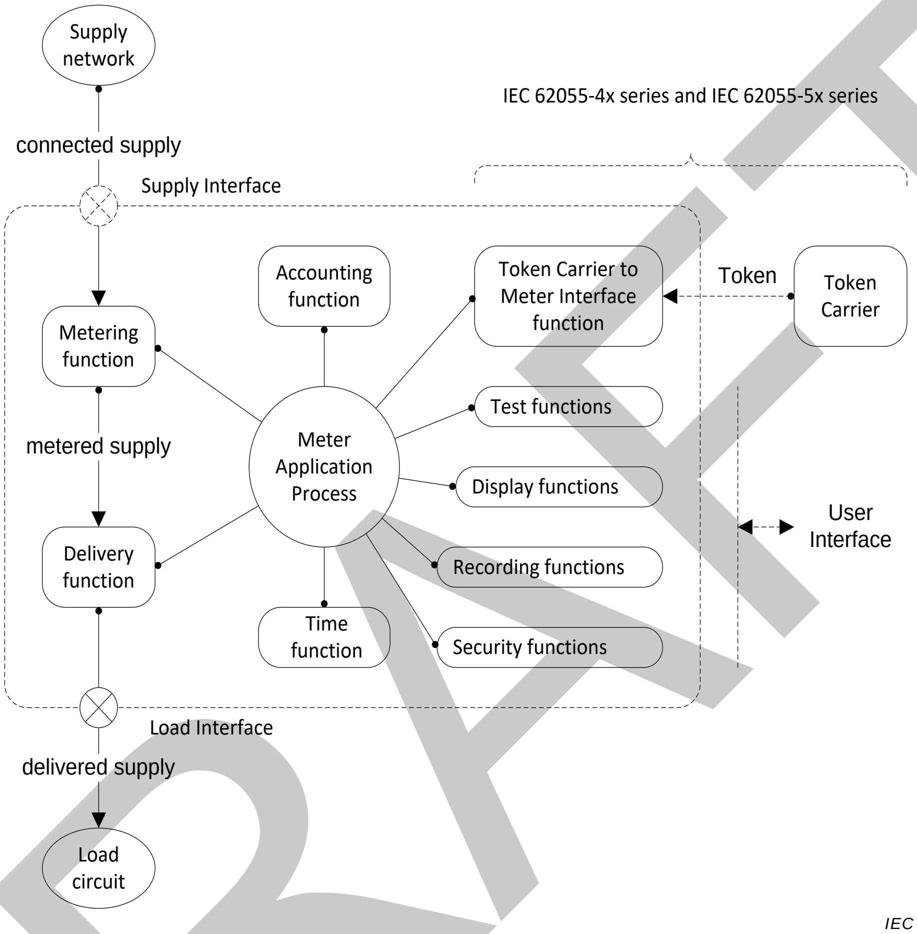


Figure 1 – Functional block diagram of a generic single-device payment meter

In a single-device payment meter all the essential functions are located in a single enclosure as depicted in Figure 1 above, while in a multi-device payment meter it is possible for the TokenCarrierToMeterInterface to be located in a separate enclosure.

The IEC 62055-4x series primarily deals with the application layer protocol and IEC 62055-5x series with the physical layer protocol of the TokenCarrierToMeterInterface. The TokenCarrier is included in the Physical Layer.

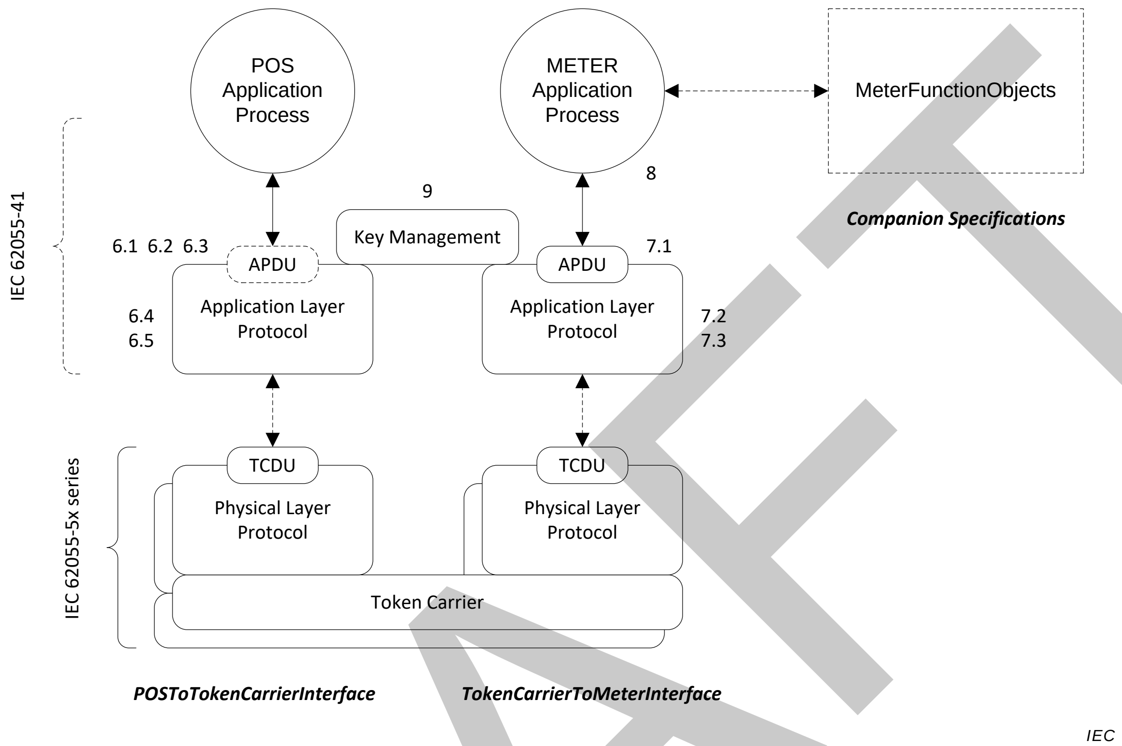
In this document the Decoder (see Clause 3) is defined as that part of the payment meter where the Application Layer functions of the TokenCarrierToMeterInterface are hosted and it is thus allocated a DRN (see 6.1.2.3).

NOTE MeterFunctionObjects are further discussed in 5.5.

In all cases, there shall only be one Application Layer implementation, thus there shall be only one DRN associated with a payment meter, whether it is a single or multi-device implementation, even though there may also be more than one Physical Layer implementation in the same payment meter.

For a more complete description of payment meter function classes see IEC TR 62055-21.

5.2 STS protocol reference model



Key

APDU ApplicationProtocolDataUnit; data interface to the application layer protocol

TCDU TokenCarrierDataUnit; data interface to the physical layer protocol

Relevant (sub)clause number references in this document are indicated adjacent to each box.

Figure 2 – STS modelled as a 2-layer collapsed OSI protocol stack

The STS is a secure data transfer protocol between a POS and a payment meter using a token carrier as the transfer medium. The application layer protocol deals with tokens and encryption processes and functions, while the physical layer protocol deals with the actual encoding of token data onto a token carrier (see Figure 2).

Examples of physically transportable token carrier devices are: numeric, magnetic cards, memory cards and memory keys. Examples of virtual token carriers are: PSTN modem, ISDN modem, GSM modem, GPRS modem, Radio modem, PLC modem, Infra-red, LAN and WAN connections and direct local connection. These are defined in the IEC 62055-5x series.

It shall be noted that although the model primarily depicts a POS to token carrier to payment meter protocol, the same protocol is equally applicable to any other device that requires communicating with the payment meter, for example CIS, MIS or portable HHU.

Although a collapsed 2-layered OSI architecture is followed in this document, it does not preclude future expansion to include more layers should the need arise or for the implementer to interpose additional layers between the two shown in this model.

The APDU is the data interface to the application layer protocol, specified in IEC 62055-41 and the TCDU is the data interface to the physical layer protocol, specified in the IEC 62055-5x series.

The STS in this document defines a one-way data transfer protocol (i.e. from POS to payment meter), although the reference model allows equally for a two-way transfer protocol, which may be a requirement in a future revision of this document.

5.3 Dataflow from the POSApplicationProcess to the TokenCarrier

The flow of data from the POSApplicationProcess to the TokenCarrier is shown in Figure 3.

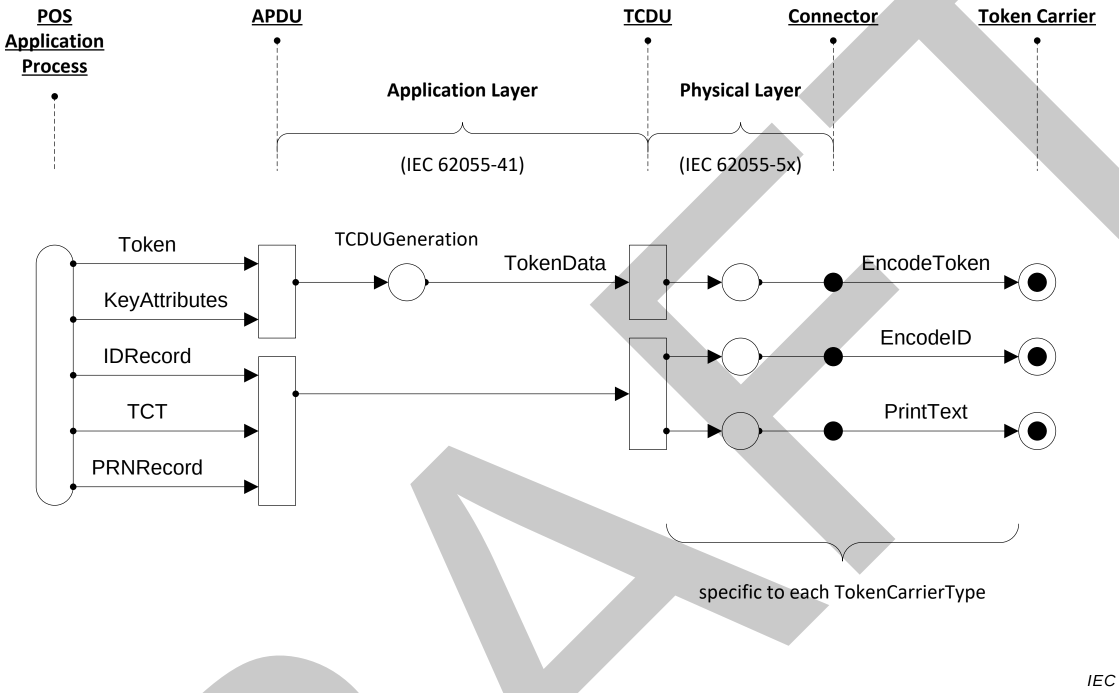


Figure 3 – Dataflow from the POSApplicationProcess to the TokenCarrier

The POSApplicationProcess presents the token to the APDU together with the KeyAttributes of the DecoderKey that is to be used for encrypting the token. The application layer protocol generates the DecoderKey, encrypts the token and presents the resultant TokenData in the TCDU. The physical layer protocol encodes the TokenData onto the TokenCarrier. Optionally, payment meter identification data may also be encoded onto the TokenCarrier (see 5.2.4 in IEC 62055-51:2007 for example) as well as printed text onto the outside surface (see 5.1.5 in IEC 62055-51:2007 for example). This part of the process essentially ends with the encoding of data onto the TokenCarrier, after which the TokenCarrier is transported to the payment meter (usually by the customer), where it is entered into the payment meter via the TokenCarrierInterface.

5.4 Dataflow from the TokenCarrier to the MeterApplicationProcess

The flow of data from the TokenCarrier to the MeterApplicationProcess is shown in Figure 4.

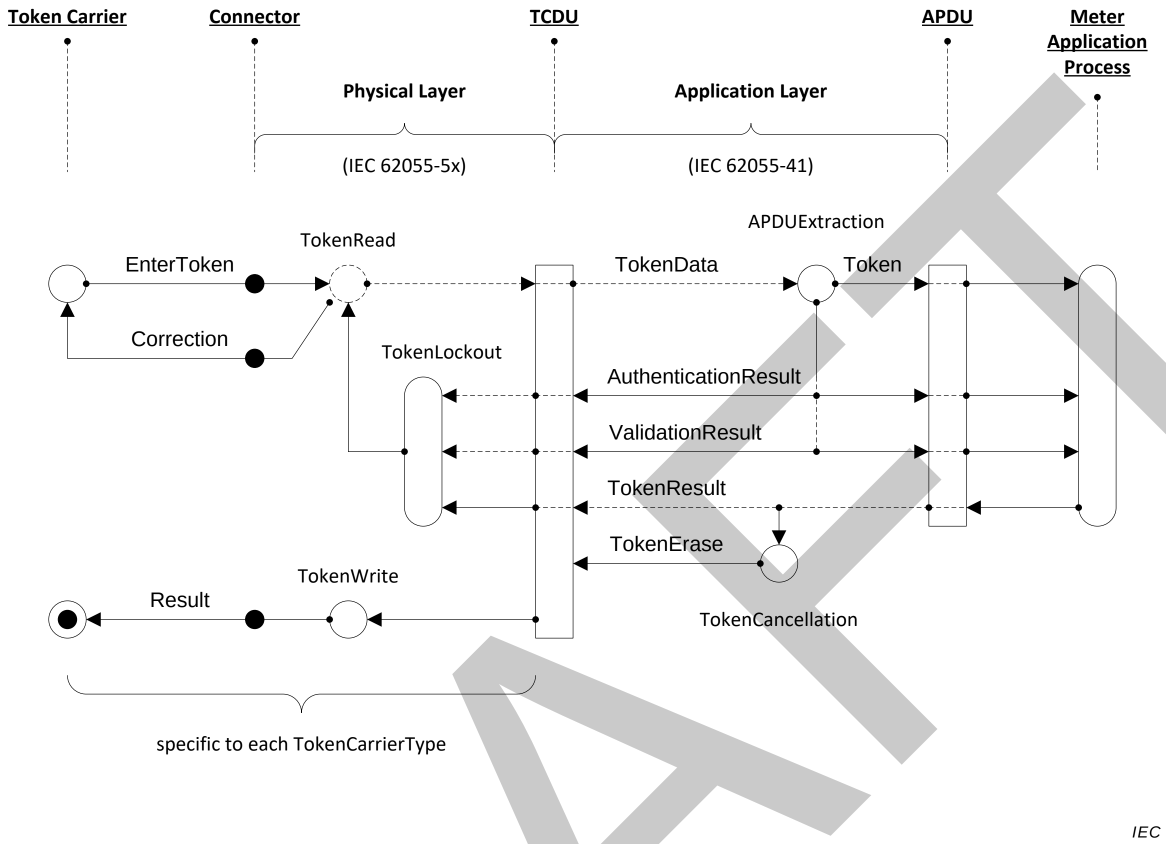


Figure 4 – Dataflow from the TokenCarrier to the MeterApplicationProcess

The token entry process from the TokenCarrier varies according to the TCT. The nature of the connector will similarly vary according to the TCT, an example of which may be a keypad or a magnetic card reader device supporting one-way token carriers as specified in IEC 62055-51.

Where other types of connectors are required to support other types of token carriers such as a memory key reader device or a plug-in connector from a hand-held unit acting as a virtual token carrier, then such token carriers shall be specified in additional parts of IEC 62055-5x in the future.

The physical layer protocol reads the token data being entered and provides immediate corrective feedback to the user (see 6.3 in IEC 62055-51:2007 for example). The entered token data is presented in the TCDU, from where the application layer protocol extracts the token by appropriate decryption, validation and authentication, the results of which are presented to the MeterApplicationProcess in the APDU. After processing and executing the instruction from the token, the MeterApplicationProcess indicates the result in the APDU for the application layer protocol to take further action. This normally causes the cancellation of the TID and the giving of the instruction, via the TCDU, to the physical layer protocol to complete the token entry process by erasure of the token data (if appropriate) or by writing of other relevant data back onto the TokenCarrier as may be appropriate.

For certain TokenCarrier types (for example a high speed virtual token carrier) the physical layer protocol may employ a token entry lockout function to protect the payment meter from fraud attempts. Typically, such a lockout function would slow down the effective rate, at which tokens may be entered via the particular token carrier interface (see 6.6.7 of IEC 62055-52:2008 for example).

DRAFT

5.5 MeterFunctionObjects / companion specifications

With reference to Figure 1 it can be seen that the TokenCarrierToMeterInterface, which also includes the TokenCarrier, is dealt with in the IEC 62055-4x and IEC 62055-5x series. The remaining MeterFunctionObjects shown in the diagram are defined in companion specifications and are not normative to this document.

Companion specifications (see Figure 2) are under the administrative control (see Clause C.9) of the STS Association and serve the purpose of defining functionality of a payment meter in a standardized way, using an object-oriented approach.

5.6 Transaction reference numbers

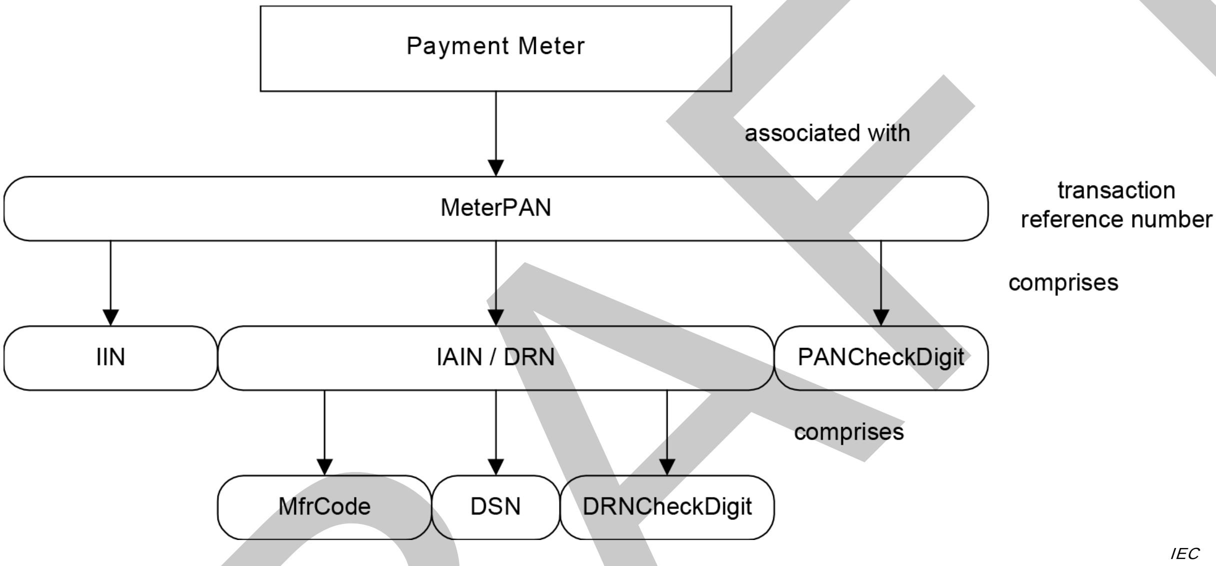


Figure 5 – Composition of transaction reference number

The transaction reference number comprises the data elements and their relationships as shown in Figure 5.

A token-based transaction (see Clause 3) constitutes a financial activity that needs to be dealt with in accordance with standard financial practices.

The PrimaryAccountNumber (PAN) serves to tag transaction records, messages, requests, authorizations and notifications, in which both transacting parties are uniquely identifiable.

A payment meter is thus uniquely associated with a MeterPAN, being a composite number comprising of IIN and IAIN / DRN, which in turn comprises MfrCode and DSN (see 6.1.2).

6 POSToTokenCarrierInterface application layer protocol

6.1 APDU: ApplicationProtocolDataUnit

6.1.1 Data elements in the APDU

The APDU is the data interface between the POSApplicationProcess and the application layer protocol and comprises the data elements given in Table 1.

Table 1 – Data elements in the APDU

Element	Context	Format	Reference
MeterPAN	Identification MeterPrimaryAccountNumber for the payment meter	18 digits	6.1.2
TCT	Directs which TokenCarrierType should be used in the physical layer protocol to carry the token to the payment meter	2 digits	6.1.3
DKGA	Directs which DecoderKeyGenerationAlgorithm is to be used for generating the DecoderKey	2 digits	6.1.4
EA	Directs which encryption algorithm is to be used for encrypting the token data	2 digits	6.1.5
SGC	Directs which SupplyGroupCode the payment meter is allocated to	6 digits	6.1.6
TI	Directs which TariffIndex the payment meter is linked to	2 digits	6.1.7
KRN	Directs which KeyRevisionNumber the DecoderKey is on (as inherited from VendingKey)	1 digit	6.1.8
KT	Directs which KeyType the DecoderKey is on	1 digit	6.1.9
KEN	A number associated with the VendingKey and a DecoderKey that determines the time period, during which the key will remain valid	8 bits	6.1.10
BaseDate	The starting date and time from which a TID is calculated	2 ASCII characters	6.1.12 6.5.3.6
Token	The actual token data that is to be transferred to the payment meter prior to encryption and processing	66 bits	6.2.1
IDRecord	Optional identification data intended to be encoded onto a payment meter ID card or onto a token carrier together with the token	35 digits	Table 2
PRNRecord	Optional print data intended to be printed at the same time as the coding of the token onto the TokenCarrier. Certain token carriers such as paper-based magnetic card devices allow printing to be done onto the card surface itself and this operation may be integrated with the magnetic card encoding device. The content and format is not specified and is left to each system to define according to its particular requirements	Undefined text	x

The optional IDRecord comprises the data elements given in Table 2.

Table 2 – Data elements in the IDRecord

Element	Context	Format	Reference
MeterPAN	Identification MeterPrimaryAccountNumber for the payment meter	18 digits	6.1.2
DOE	Optional expiry date for the identification data as encoded onto a payment meter ID card or token carrier (as an example, see IEC 62055-51)	4 digits	6.1.11
TCT	Indicates which TokenCarrierType is associated with this MeterPAN	2 digits	6.1.3
EA	Indicates which encryption algorithm is associated with this MeterPAN	2 digits	6.1.5
SGC	Indicates which SupplyGroupCode is associated with this MeterPAN	6 digits	6.1.6
TI	Indicates which TariffIndex is associated with this MeterPAN	2 digits	6.1.7
KRN	Indicates which KeyRevisionNumber is associated with this MeterPAN (as inherited from VendingKey)	1 digit	6.1.8

6.1.2 MeterPAN: MeterPrimaryAccountNumber

6.1.2.1 Data elements in the MeterPAN

The MeterPAN is a unique identification number for each STS-compliant payment meter. It comprises the 3 parts given in Table 3.

Table 3 – Data elements in the MeterPAN

Element	Context	Format	Reference
IIN	IssuerIdentificationNumber	4/6 digits	6.1.2.2
IAIN / DRN	IndividualAccountIdentificationNumber / DecoderReferenceNumber	11/13 digits	6.1.2.3
PANCheckDigit	Result of a formula to check the integrity of the IIN and the IAIN	1 digit	6.1.2.4
NOTE The first digit of the IIN is the most significant digit of the 18-digit MeterPAN and the PANCheckDigit is the least significant digit.			

See also Annex C for Code of practice on managing this data element.

6.1.2.2 IIN: IssuerIdentificationNumber

The IIN is a unique 6/4-digit number that defines a domain, under which further IAIN values (i.e. DRN values) may be issued for use within this defined domain.

For 11-digit DRNs the IIN shall be 600727 and for 13digit DRNs the IIN shall be 0000.

See also C.4.2 on managing this data element.

6.1.2.3 IAIN: IndividualAccountIdentificationNumber/ DRN: DecoderReferenceNumber

6.1.2.3.1 Data elements in the IAIN / DRN

A unique DRN shall be allocated to the device that performs the application layer protocol in an STS-compliant payment meter.

NOTE In many systems, the decoder part is integral with the metering part and hence the DRN might be synonymous with the meter serial number.

The DRN is an 11/13-digit number comprising of the data elements given in Table 4.

Table 4 – Data elements in the IAIN / DRN

Element	Context	Format	Reference
MfrCode	A number to uniquely identify a payment meter manufacturer	2/4 digits	6.1.2.3.2
DSN	An eight digit serial number allocated by the manufacturer	8 digits	6.1.2.3.3
DRNCheckDigit	Check Digit; formula to check the integrity of the MfrCode and the DSN	1 digit	6.1.2.3.4
NOTE The MfrCode is the 2/4 most significant digits of the 11/13-digit DRN and the DRNCheckDigit is the least significant digit.			

MfrCode values shall always be right justified and left padded with 0.

The DSN shall be right justified and left padded with 0 to a full 8-digit string.

6.1.2.3.2 MfrCode: ManufacturerCode

The MfrCode is a 2/4-digit number that shall be used to uniquely identify the manufacturer of the payment meter.

The STS Association provides a service for the allocation of MfrCode values to uniquely identify manufacturers in order to ensure interoperability of STS-compliant equipment.

MfrCode values 00 and 0100 are reserved for product certification test purposes and shall not be used in any production equipment.

See also C.4.3 on managing this data element.

6.1.2.3.3 DSN: DecoderSerialNumber

The DSN is a unique 8-digit serial number that is generated internally by the manufacturer. Each manufacturer is responsible for the uniqueness of the DSN with respect to his MfrCode.

See also C.4.4 on managing this data element.

6.1.2.3.4 DRNCheckDigit

The DRNCheckDigit is a single digit used to validate the integrity of the MfrCode and DSN values when being entered by hand or being read by machine. This is a modulus 10 check digit, calculated using the Luhn formula, as illustrated in Annex B of ISO/IEC 7812-1:2006. It is calculated on the 10/12 preceding digits of the DRN generated through the concatenation of the MfrCode and the DSN values.

6.1.2.4 PANCheckDigit

The PANCheckDigit is a single digit used to validate the integrity of the IIN and the IAIN values when being entered by hand or being read by machine. The method used to calculate the PANCheckDigit value is given in 4.4 of ISO/IEC 7812-1:2006 and is calculated on the preceding 17 digits of the MeterPAN generated through the concatenation of the IIN and the IAIN values.

6.1.3 TCT: TokenCarrierType

This is a 2-digit number used to uniquely identify the type of token carrier onto which the token should be encoded for transferring to the payment meter. The values for token carrier types are given in Table 5.

Table 5 – Token carrier types

Code	TokenCarrier	Comments
00	Reserved	For future assignment by the STS Association
01	Magnetic card	As defined in IEC 62055-51
02	Numeric	As defined in IEC 62055-51
03-06	Reserved	Legacy systems using proprietary token carrier technologies
07	Virtual Token Carrier (VTC07)	As defined in IEC 62055-52
08	DLMS_COSEM_VTC (VTC08)	Virtual token carrier type for transporting STS tokens over DLMS/COSEM
09-99	Reserved	For future assignment by the STS Association
NOTE TCT08 is provisioned for a future standard.		

Values less than 10 shall be right justified and left padded with 0 (for example 01, 02-09).

6.1.4 DKGA: DecoderKeyGenerationAlgorithm

This is a 2-digit number used to uniquely identify which algorithm is to be used for generating the DecoderKey. The DKGA code values are given in Table 6.

Table 6 – DKGA codes

Code	DKG algorithm	Comments	Reference
00	Reserved	For future assignment by the STS Association	x
01	DKGA01	Limited number of early legacy STS-compliant payment meters. Superseded by DKGA02	6.5.3.3
02	DKGA02	System using 64-bit DES VendingKey derivation	6.5.3.4
03	DKGA03	System using dual 64-bit DES VendingKey derivation	6.5.3.5
04	DKGA04	System using KDF-HMAC-SHA-256 VendingKey derivation	6.5.3.6
05-99	Reserved	For future assignment by the STS Association	x
DKGA02 is the algorithm to be used for current systems, subject to the criteria for DKGA01. DKGA03 is deprecated and shall not be used for new products. DKGA04 shall be deployed in advance of, or in conjunction with, the introduction of meters using EA code 07 or code 11. See also 6.1.5.			

Values less than 10 shall be right justified and left padded with 0 (for example 01, 02-09).

6.1.5 EA: EncryptionAlgorithm

This is a 2-digit number used to uniquely identify which algorithm is to be used for encrypting the token data. The EA code values are given in Table 7.

Table 7 – EA codes

Code	EncryptionAlgorithm	Comments	Reference
00	Reserved	For future assignment by the STS Association	x
01-06	Reserved	Legacy proprietary systems	x
07	STA	Systems using the Standard Transfer Algorithm as defined in this document	6.5.4.1
08	Reserved	Legacy proprietary systems	x
09	DEA	Systems using the Data Encryption Algorithm as defined in ANSI X3.92	6.5.5
10	Reserved	Legacy proprietary systems	x
11	MISTY1	Systems using the Encryption Algorithm as defined in ISO/IEC 18033-3 as for MISTY1	6.5.6
12-99	Reserved	For future assignment by the STS Association	x
EA09 is deprecated and shall not be used for new products.			

Values less than 10 shall be right justified and left padded with 0. For example 01, 02-09.

6.1.6 SGC: SupplyGroupCode

This is a unique 6-digit decimal number allocated to a utility, which is registered within the KMS. It is used to uniquely identify a sub-group of payment meters within the supply or distribution domain of the utility. Each SupplyGroup has one or more VendingKeys associated with it. Each payment meter in the SupplyGroup has a DecoderKey derived from one of these VendingKeys.

DRAFT

Token sales authorisation is thus controlled by selective distribution of such VendingKey and SGC to authorised token vendor agents operating POS services on behalf of utilities. SGC management and VendingKey management is completely under the control of the KMS and is subject to such Code of practice.

Values less than 6 decimal digits shall be right justified and left padded with 0. For example 000001, 000002.. 000009.

The SGC inherits its type from the KT attribute of the VendingKey (see 6.5.2.2.1), to which it is associated as shown in Table 8. A particular SGC may inherit more than one KT at the same time during the operational life of the SGC.

Table 8 – SGC types and key types

KT	SGC type	VendingKey type (see 6.5.2.2.1)	DecoderKey type (see 6.5.2.3.1)
0	Initialization	Not specified	DITK
1	Default	VDDK	DDTK
2	Unique	VUDK	DUTK
3	Common	VCDK	DCTK

See also C.3.2 for Code of practice on managing this data element.

6.1.7 TI: TariffIndex

A 2-digit number associated with a particular tariff that is allocated to a particular customer. The maintenance and the content of the tariff tables are the responsibility of the utility.

Values less than 10 shall be right justified and left padded with 0 (for example 01, 02.. 09).

The TI is also encoded into the DecoderKey, which means that when a customer is moved from one TI to another, then his DecoderKey will also have to change (see 6.5.2.1).

NOTE The encoding of this value when used in the ControlBlock for Decoder Key Generation (see 6.5.3.2) is as two hexadecimal digits, whereas the encoding as used in the Set2ndSectionDecoderKey token (see 6.2.7.3) is as an 8 bit binary number. In these cases a tariff index of 99 decimal is encoded as binary string 10011001 and 0110 0011 respectively.

See also Clause C.10 for Code of practice on managing this data element.

6.1.8 KRN: KeyRevisionNumber

This is a 1-digit number in the range 1 to 9, which uniquely identifies a VendingKey within a SupplyGroup. A payment meter’s DecoderKey is associated with the SGC and KRN of the VendingKey from which it is derived.

See 6.5.2.5 for a detailed definition of this data element.

6.1.9 KT: KeyType

This is a 1-digit number in the range 0 to 3 associated with a property of the VendingKey and thus also with the corresponding DecoderKey, which is derived from the VendingKey.

See 6.5.2 for a detailed definition of this data element.

6.1.10 KEN: KeyExpiryNumber

A KEN is associated with each VendingKey by the KMS, and defines the time when a VendingKey and any corresponding DecoderKey will expire, after which it becomes invalid for further use, subject to certain concessions.

The KEN corresponds to the most significant 8 bits of the 24-bit TID. Any token identifier whose most significant 8 bits are greater than a given key's KEN cannot be encrypted or decrypted with that key.

See 6.5.2.6 for a detailed definition of this data element.

See also C.3.4 for Code of practice on managing this data element.

6.1.11 DOE: DateOfExpiry

The use of this date is optional and is associated with a validity period for identity related data that gets encoded onto an identity-carrying device. For example: a payment meter ID card or a second record encoded onto the TokenCarrier together with the token data. In some implementations it is found to be useful to let the customer bring back a used token carrier to serve as his decoder identification to the POS when purchasing his next token. (See for example 5.1.4 and 5.2.4.9 of IEC 62055-51:2007).

This date may also be used, for example, in cases where a consumer has been granted a concessionary tariff for a limited period. The date encoded is the last month for which the card is valid.

DOE is in the format YYMM and shall always contain 4 digits.

Where YY or MM is less than 10, it shall be right justified and left padded with 0 (for example 01, 02, 09, etc.).

When the DOE in the IDRecord is not used, then YYMM = 0000.

DOE code values for the year and month are given in Table 9 and Table 10.

Table 9 – DOE codes for the year

YY	Represents
00	2000 or DOE is not used (see also Table 10)
01 – 78	2001 – 2078

Table 10 – DOE codes for the month

MM	Represents
00	DOE is not used (see also Table 9)
01 – 12	Jan – Dec
13 – 99	Invalid

6.1.12 BDT: BaseDate

The BaseDate is a date and time marker, from which a token identifier (TID) is calculated (see 6.3.5 for using the BaseDate to calculate a TID).

DRAFT

BaseDate is given with respect to Coordinated Universal Time (UTC) time zone.

In order to accommodate the fact that the 24-bit TID will roll over approximately every 31 years, three BaseDate values are defined and are given in Table 11.

Table 11 – BDT representation

Date	BDT representation
01 January 1993, 00:00:00 UTC	93
01 January 2014, 00:00:00 UTC	14
01 January 2035, 00:00:00 UTC	35

6.2 Tokens

6.2.1 Token definition format

The TokenData element in the APDU is a 66-bit binary number comprising of several fields of smaller data elements, in accordance with which various processes are initiated in the MeterApplicationProcess and various bits of information are transferred to the payment meter registers.

The definition format for the tokens in 6.2.2 to 6.2.14 is given in Table 12.

Table 12 – Token definition format

Name of data element	Example:	Class, SubClass, RND, TID, Amount, CRC, etc.
Number of bits	Example:	2 bits, 4 bits, 24 bits, 16 bits, etc.
Range of values	Example:	1, 2, 5-15, etc.

6.2.2 Class 0: TransferCredit

Class	**SubClass**	**RND**	**TID**	**Amount**	**CRC**
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
0	0 = electricity 1 = water 2 = gas 3 = time				

Class	**SubClass**	**S&E**	**TID**	**Amount**	**CRC_C**
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
0	4 = electricity currency 5 = water currency 6 = gas currency 7 = time currency 8-15 = future assignment				

Action: Transfer credit to the payment meter to the value as defined in the Amount field (see 6.3.6) and for the service type as defined in the SubClass field.

6.2.3 Class 1: InitiateMeterTest/Display

Class	SubClass	Control	MfrCode	CRC
2 bits	**4 bits**	**36/28 bits**	**8/16 bits**	**16 bits**
1	0 = STS defined	Bit position control of test/display number for 2 digit manufacturer codes. Use 36 bits.	0 (8 bits)	
1	1 = STS defined	Bit position control of test/display number for 4 digit manufacturer codes. Use 28 bits	0 (16 bits)	
1	2-5 = reserved for future assignment by the STS Association.	Reserved for future assignment by the STS Association.	Reserved for future assignment by the STS Association.	
1	6-10 = proprietary use.	For 4 digit manufacturer codes. If not used, set to zero (28 bits)	0100-9999 (16 bits)	
1	11-15 = proprietary use	For 2 digit manufacturer codes. If not used, set to zero (36 bits)	00-99 (8 bits)	

Action: Initiate the test or display function in the payment meter in accordance with the bit pattern defined in the Control field (see 6.3.8).

A meter having a 2-digit MfrCode value shall support the 36-bit Control field format and may also optionally support the 28-bit Control field format.

A meter having a 4-digit MfrCode value shall support the 28-bit Control field format and may also optionally support the 36-bit Control field format.

6.2.4 Class 2: SetMaximumPowerLimit

Class	SubClass	RND	TID	MPL	CRC
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
2	0				

Action: Load the maximum power limit register in the payment meter with the value as given in the MPL field (see 6.3.9).

6.2.5 Class 2: ClearCredit

Class	SubClass	RND	TID	Register	CRC
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
2	1				

Action: Clear the corresponding credit register as indicated in the Register field (see 6.3.13) in the payment meter to zero.

6.2.6 Class 2: SetTariffRate

Class	SubClass	RND	TID	Rate	CRC
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
2	2				

DRAFT

Action: Load the tariff rate register in the payment meter with the value given in the Rate field (see 6.3.11).

This token is reserved for future definition by the STS Association.

6.2.7 Key change token set for 64-bit DecoderKey transfer

6.2.7.1 General

For 64-bit DecoderKey transfers the decoder shall support a two-token set and optionally a three-token set.

The two-token set shall comprise of the following tokens:

- Set1stSectionDecoderKey;
- Set2ndSectionDecoderKey.

The three-token set shall comprise of the following tokens:

- Set1stSectionDecoderKey token;
- Set2ndSectionDecoderKey token;
- Set3rdSectionDecoderKey token.

6.2.7.2 Class 2: Set1stSectionDecoderKey

Class	SubClass	KENHO	KRN	RO	3KCT	KT	NKHO	CRC
2 bits	**4 bits**	**4 bits**	**4 bits**	**1 bit**	**1 bit**	**2 bits**	**32 bits**	**16 bits**
2	3		1-9	0-1	0-1	0-3		

Action: Load the DecoderKeyRegister with the 1st half of the new DecoderKey. See 8.9 for the processing of this token.

For decoders that support the three-token set the 3KCT field shall be set to 1 if Set3rdSectionDecoderKey token is included in the set. It shall be set to 0 if Set3rdSectionDecoderKey token is not included in the set.

6.2.7.3 Class 2: Set2ndSectionDecoderKey

Class	SubClass	KENLO	TI	NKLO	CRC
2 bits	**4 bits**	**4 bits**	**8 bits**	**32 bits**	**16 bits**
2	4		0-99		

Action: Load the DecoderKeyRegister with the 2nd half of the new DecoderKey. See 8.9 for the processing of this token.

6.2.7.4 Class 2: Set3rdSectionDecoderKey

Class	SubClass	SGC	Res_A	CRC
2 bits	**4 bits**	**24 bits**	**20 bits**	**16 bits**
2	8	0-999999	0	

NOTE The SGC values 1000000 – 16777215 are for future assignment by the STS Association.

The Res_A reserved bits shall be set to 0.

DRAFT

Action: Load the DecoderKeyRegister with the SGC of the new DecoderKey. See 8.9 for the processing of this token.

6.2.8 Key change token set for 128-bit DecoderKey transfer

6.2.8.1 General

For 128-bit DecoderKey transfers the decoder shall support a four-token set.

The four-token set shall comprise of the following tokens:

- Set1stSectionDecoderKey;
- Set2ndSectionDecoderKey;
- Set3rdSectionDecoderKey;
- Set4thSectionDecoderKey.

The DecoderKey = concatenate(NKHO, NKMO2, NKMO1, NKLO).

The SGC = concatenate(SGCHO, SGCLO).

6.2.8.2 Class 2: Set1stSectionDecoderKey

Class	SubClass	KENHO	KRN	RO	Res_B	KT	NKHO	CRC
2 bits	**4 bits**	**4 bits**	**4 bits**	**1 bit**	**1 bit**	**2 bits**	**32 bits**	**16 bits**
2	3		1-9	0-1	0	0-3		
The Res_B reserved bit shall be set to 0.								

Action: Transfer the NKHO bits of the new DecoderKey to the decoder. See 8.9 for the processing of this token.

6.2.8.3 Class 2: Set2ndSectionDecoderKey

Class	SubClass	KENLO	TI	NKLO	CRC
2 bits	**4 bits**	**4 bits**	**8 bits**	**32 bits**	**16 bits**
2	4		0-99		

Action: Transfer the NKLO bits of the new DecoderKey to the decoder. See 8.9 for the processing of this token.

6.2.8.4 Class 2: Set3rdSectionDecoderKey

Class	SubClass	SGCLO	NKMO2	CRC
2 bits	**4 bits**	**12 bits**	**32 bits**	**16 bits**
2	8			

Action: Transfer the NKMO2 bits of the new DecoderKey to the decoder. See 8.9 for the processing of this token.

6.2.8.5 Class 2: Set4thSectionDecoderKey

Class	SubClass	SGCHO	NKMO1	CRC
2 bits	**4 bits**	**12 bits**	**32 bits**	**16 bits**
2	9			

Action: Transfer the NKMO1 bits of the new DecoderKey to the decoder. See 8.9 for the processing of this token.

6.2.9 Class 2: ClearTamperCondition

Class	SubClass	RND	TID	Pad	CRC
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
2	5			0	

Action: Clear the tamper status register in the payment meter and cancel any resultant control processes that may be in progress due to the tamper condition.

6.2.10 Class 2: SetMaximumPhasePowerUnbalanceLimit

Class	SubClass	RND	TID	MPPUL	CRC
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
2	6				

Action: Load the maximum phase unbalance limit register in the payment meter with the value given in the MPPUL field (see 6.3.10). See also 8.12 for more detail on the action of this function in the payment meter.

6.2.11 Class 2: SetWaterMeterFactor

Class	SubClass	RND	TID	WMFactor	CRC
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
2	7				

Action: Load the water meter factor register in the payment meter with the value given in the WMFactor field (see 6.3.12).

This token is reserved by the STS Association for water applications.

6.2.12 Class 2: Reserved for STS use

Class	SubClass	RND	TID	ResData	CRC
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
2	10				

Action: Reserved for future definition by the STS Association.

This token range is reserved for future assignment by the STS Association.

6.2.13 Class 2: Reserved for Proprietary use

Class	SubClass	RND	TID	PropData	CRC
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
2	11-15				

Action: Defined by manufacturer.

This token range is reserved for proprietary definition and use.

This document does not provide protection against collision between manufacturer uses of this token space. Generation and control of these tokens shall therefore always be under the direct management of the relevant manufacturer and shall never be available on vending systems for general use within STS-compliant payment metering systems.

6.2.14 Class 3: Reserved for STS use

Class	SubClass	Res_B
2 bits	**4 bits**	**60 bits**
3	0-15	

Action: Reserved for future definition by the STS Association.

This token range is reserved for future assignment by the STS Association.

6.3 Token data elements

6.3.1 Data elements used in tokens

The data elements given in Table 13 are used in tokens in various combinations and are all encoded in binary format.

Table 13 – Data elements used in tokens

Element	Name	Format	Reference
3KCT	TripletKeyChangeTokenFlag (see also 6.2.7.2)	1 bit	
Amount	TransferAmount (see also 6.2.2)	16 bits	6.3.6
Class	TokenClass (see also 6.2.2 to 6.2.14)	2 bits	6.3.2
Control	InitiateMeterTest/DisplayControlField (see also 6.2.3)	36/28 bits	6.3.8
CRC	CyclicRedundancyCheck (see also 6.2.2 to 6.2.13)	16 bits	6.3.7
CRC_C	CyclicRedundancyCheck_C (see also 6.2.2)	16 bits	6.3.22
KENHO	KeyExpiryNumberHighOrder (see also 6.2.7)	4 bits	6.3.18
KENLO	KeyExpiryNumberLowOrder (see also 6.2.7.3)	4 bits	6.3.19
KRN	KeyRevisionNumber (see also 6.2.7)	4 bits	6.1.8
KT	KeyType (see also 6.2.7)	2 bits	6.1.9
MfrCode	ManufacturerCode (see also 6.2.3)	8/16 bits	6.1.2.3.2
MPL	MaximumPowerLimit (see also 6.2.4)	16 bits	6.3.9
MPPUL	MaximumPhasePowerUnbalanceLimit (see also 6.2.10)	16 bits	6.3.10
NKHO	NewKeyHighOrder (see also 6.2.7)	32 bits	6.3.14
NKLO	NewKeyLowOrder (see also 6.2.7.3)	32 bits	6.3.15

DRAFT

Element	Name	Format	Reference
NKMO1	NewKeyMiddleOrder1 (see also 6.2.8.5)	32 bits	
NKMO2	NewKeyMiddleOrder2 (see also 6.2.8.4)	32 bits	
Pad	Pad value with 0 (see also 6.2.9)	16 bits	x
PropData	Proprietary data field (see also 6.2.13)	16 bits	x
Rate	[TariffRate] For future definition (see also 6.2.6)	16 bits	6.3.11
Register	RegisterToClear (see also 6.2.5)	16 bits	6.3.13
Res_A	Reserved for future assignment (see also 6.2.7.4)	20 bits	x
Res_B	Reserved for future assignment (see also 6.2.8.2 and 6.2.14)	1 bits	x
ResData	Reserved data field for future assignment (see also 6.2.12)	16 bits	x
RND	RandomNumber (see also 6.2.2 to 6.2.13)	4 bits	6.3.4
RO	RolloverKeyChange (see also 6.2.7)	1 bits	6.3.20
SGC	SupplyGroupCode (see also 6.2.8)	24 bits	6.1.6
SGCHO	SupplyGroupCodeHighOrder	12 bits	
SGCLO	SupplyGroupCodeLowOrder	12 bits	
SubClass	TokenSubClass (see also 6.2.2 to 6.2.14)	4 bits	6.3.3
S&E	SignAndExponent (see also 6.2.2)	4 bits	6.3.21
TI	TariffIndex (see also 6.2.7.3)	8 bits	6.1.7
TID	TokenIdentifier (see also 6.2.2 to 6.2.13)	24 bits	6.3.5.1
WMFactor	[WaterMeterFactor] Reserved by the STS Association for water application (see also 6.2.11)	16 bits	6.3.12

6.3.2 Class: TokenClass

Tokens are classified into 4 main functional areas as given in Table 14.

Table 14 – Token classes

TokenClass	Function
0	Credit transfer
1	Non-meter-specific management
2	Meter-specific management
3	Reserved for future assignment by the STS Association

Class 0 and Class 2 tokens are encrypted using the DecoderKey, while Class 1 tokens are not encrypted and can thus be used on any STS-compliant payment meter.

DRAFT

6.3.3 SubClass: TokenSubClass

Further sub-classification of the TokenClass is given in Table 15.

Table 15 – Token sub-classes

<table>
<tr><th rowspan="2">Token SubClass</th><th colspan="4">Token Class</th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>0</td><td>TransferCredit (electricity)</td><td>InitiateMeterTest/Display for 2-digit MfrCode</td><td>SetMaximumPowerLimit</td><td rowspan="16">Reserved for future assignment by the STS Association</td></tr>
<tr><td>1</td><td>TransferCredit (water)</td><td>InitiateMeterTest/Display for 4-digit MfrCode</td><td>ClearCredit</td></tr>
<tr><td>2</td><td>TransferCredit (gas)</td><td rowspan="4">Reserved for future assignment by the STS Association</td><td>SetTariffRate
Reserved for future assignment by the STS Association</td></tr>
<tr><td>3</td><td>TransferCredit (time)</td><td>Set1stSectionDecoderKey</td></tr>
<tr><td>4</td><td>TransferCredit (electricity currency)</td><td>Set2ndSectionDecoderKey</td></tr>
<tr><td>5</td><td>TransferCredit (water currency)</td><td>ClearTamperCondition</td></tr>
<tr><td>6</td><td>TransferCredit (gas currency)</td><td rowspan="5">Reserved for proprietary use for 4-digit MfrCode</td><td>SetMaximumPhasePower
UnbalanceLimit</td></tr>
<tr><td>7</td><td>TransferCredit (time currency)</td><td>SetWaterMeterFactor
Reserved by the STS Association for future assignment</td></tr>
<tr><td>8</td><td rowspan="8">Reserved for future assignment by the STS Association</td><td>Set3rdSectionDecoderKey</td></tr>
<tr><td>9</td><td>Set4thSectionDecoderKey</td></tr>
<tr><td>10</td><td>Reserved for future assignment by the STS Association</td></tr>
<tr><td>11</td><td rowspan="5">Reserved for proprietary use for 2-digit MfrCode</td><td rowspan="5">Reserved for proprietary use</td></tr>
<tr><td>12</td></tr>
<tr><td>13</td></tr>
<tr><td>14</td></tr>
<tr><td>15</td></tr>
</table>

6.3.4 RND: RandomNumber

The generation of this 4-bit number will be a snapshot of the four least significant bits of at least a millisecond counter. The inclusion of a random number in the data to be transferred enhances the security of the token transfer by providing a probability of 16:1 that no two tokens containing identical data to be transferred will have the same binary pattern. The control of this data element shall be implemented in a secure environment such as a hardware cryptographic module.

DRAFT

6.3.5 TID: TokenIdentifier

6.3.5.1 TID calculation

The TID field is derived from the date and time of issue and indicates the number of minutes elapsed from the BaseDate associated with the VendingKey. This field is a 24-bit binary representation of the elapsed minutes.

NOTE The definition of BaseDate now references UTC (see 6.1.12), whereas previously it implicitly referenced local time.

For example: with a date and time format of YYYY:MM:DD:hh:mm:ss the BaseDate and time of 1993:01:01:00:00:00 corresponds to a TID value of 0.

The calculation of elapsed minutes shall take leap years into account.

The rule used to determine a leap year is:

- the month of February shall have an extra day in all years that are evenly divisible by 4, except for century years (those ending in -00), which receive the extra day only if they are evenly divisible by 400. Thus 1996 was a leap year whereas 1999 was not, and 1600, 2000 and 2400 are leap years but 1700, 1800, 1900 and 2100 are not.

In the binary representation of the TID the leftmost bit represents the most significant bit.

When calculating the TID the ":ss" value shall be truncated from the actual time.

Examples of TID calculated values are given in Table 16.

Table 16 – TID calculation examples

BDT	Date of issue	Time of issue	Elapsed minutes	Resultant 24-bit TID
93	1 January 1993	00:00:00	0	0000 0000 0000 0000 0000 0000
93	1 January 1993	00:01:45	1	0000 0000 0000 0000 0000 0001
93	25 March 1993	13:55:22	120,355	0000 0001 1101 0110 0010 0011
93	25 March 1996	13:55:22	1,698,595	0001 1001 1110 1011 0010 0011
93	1 November 2005	00:01:55	6,749,281	0110 0110 1111 1100 0110 0001
93	1 December 2015	00:01:05	12,051,361	1011 0111 1110 0011 1010 0001
93	24 November 2024	20:15:00	16,777,215	1111 1111 1111 1111 1111 1111
14	1 January 2014	00:00:00	0	0000 0000 0000 0000 0000 0000
14	24 November 2045	20:15:00	16,777,215	1111 1111 1111 1111 1111 1111
35	1 January 2035	00:00:00	0	0000 0000 0000 0000 0000 0000
35	24 November 2066	20:15:00	16,777,215	1111 1111 1111 1111 1111 1111

In order to prevent token re-use when a BaseDate change is performed, certain operational procedures need to be performed. Refer to Clause C.12 for additional information.

6.3.5.2 SpecialReservedTokenIdentifier

The TokenIdentifier corresponding to 00 h 01 min of each day is reserved for special application tokens and may not be used for any other token.

Using the date and time format of YYYY:MM:DD:hh:mm:ss the reserved TID values correspond to xxxx:xx:xx:00:01:xx.

DRAFT

If a token, other than a special application token is to be generated on a time corresponding to this reserved TID, then 1 min shall be added to the TID.

See also Clause C.5 Code of practice for the management of this special reserved TID.

The use of special application tokens are optional (see Clause C.12), but the rule for how to use the special reserved TID is mandatory.

6.3.5.3 Multiple tokens generated within the same minute

The POS shall ensure that no legitimately purchased token can carry the same TID as that of any other legitimately purchased token for the same payment meter even if more than one token is purchased within the same minute on the same POS.

If multiple tokens need to be generated within the same minute for the same payment meter, then 1 min shall be added to the TID of each successive token in the set. At the end of the token generating process the POS shall revert back to real time again.

This shall apply to any token that implements a TID.

This shall not apply to special application tokens that implement the SpecialReserved TokenIdentifier (see 6.3.5.2).

For example: if 3 credit tokens A, B and C are generated within the same minute at 13h23 and in sequential order A, B and C, then A shall carry the TID time stamp 13h23, B shall carry time stamp 13h24 and C shall carry 13h25.

6.3.6 Amount: TransferAmount

6.3.6.1 General

TransferAmount is the amount of service units or currency units coded into the Amount field of the token and received by the meter.

The associated unit for the TransferAmount is defined in Table 17.

Table 17 – Units of measure for electricity

Transfer type	Units of measure
Electrical energy	watt-hours × 100 (0,1 kWh)
Electrical power	watts
Electrical currency	10^{-5} base currency

DRAFT

The STS Association also reserves the transfer types given in Table 18 for other applications.

Table 18 – Units of measure for other applications

Transfer type	Units of measure
Water	0,1 cubic metres
Gas	0,1 cubic metres
Time	0,1 minutes
Water currency	10^{-5} base currency
Gas currency	10^{-5} base currency
Time currency	10^{-5} base currency
NOTE The STS Association reserves the right to define other future transfer types for other utility services.	

6.3.6.2 Amount for SubClass 0 to 3

The 16 bits of the Amount field are subdivided into two sections, a base-10 exponent of 2 bits and a mantissa of 14 bits. The bits are numbered from right to left, starting at 0. Bit 15 is the most significant bit of the exponent and Bit 13 is the most significant bit of the mantissa. The bit allocations within this field are illustrated in Table 19.

Table 19 – Bit allocations for the Amount field for SubClass 0 to 3

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	e	e	m	m	m	m	m	m	m	m	m	m	m	m	m	m

The mathematical formula for TransferAmount conversion is as follows:

$t = 10^{e} \times m$, for $e = 0$

or

$$t = (10^{e} \times m) + \sum_{n=1}^{e}\left(2^{14} \times 10^{(n-1)}\right), \text{ for } e > 0$$

where:

t is the TransferAmount;

e is the base 10 exponent;

m is the mantissa; and

n is an integer in the range 1 to e inclusive.

All TransferAmount conversions shall be rounded up in favour of the customer. The possible TransferAmount ranges and the associated maximum errors that can arise owing to rounding up are shown in Table 20. Examples of TransferAmount values are given in Table 21.

DRAFT

Table 20 – Maximum error due to rounding

Exponent value	TransferAmount range	Maximum error
0	0000000 to 00016383	0,000
1	0016384 to 00180214	0,055 %
2	0180224 to 01818524	0,055 %
3	1818624 to 18201624	0,055 %

Table 21 – Examples of TransferAmount values for credit transfer

Item	Units purchased	Resultant 16-bit Amount field	TransferAmount Units converted and received by the meter
1	0,1 kWh	0000 0000 0000 0001	0,1 kWh
2	25,6 kWh	0000 0001 0000 0000	25,6 kWh
3	1638,3 kWh	0011 1111 1111 1111	1638,3 kWh
4	1638,4 kWh	0100 0000 0000 0000	1 638,4 kWh
5	18022,3 kWh	0111 1111 1111 1111	18022,4 kWh
6	18022,4 kWh	1000 0000 0000 0000	18022,4 kWh
7	181862,3 kWh	1011 1111 1111 1111	181862,4 kWh
8	181862,4 kWh	1100 0000 0000 0000	181862,4 kWh
9	1820162,4 kWh	1111 1111 1111 1111	1820162,4 kWh

6.3.6.3 Amount for SubClass 4 to 7

The bit allocation for Amount field is given in Table 22.

Table 22 – Bit allocations for the Amount field for SubClass 4 to 7

Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit value	e_1	e_0	m	m	m	m	m	m	m	m	m	m	m	m	m	m

The final value of e is calculated from e_4, e_3, e_2, e_1 and e_0, obtained from 6.3.21, Table 29 and Table 22 and assigning them bit values as given in Table 23.

Table 23 – Bit allocations for the exponent *e*

Bit position	4	3	2	1	0
Bit value	e_4	e_3	e_2	e_1	e_0

$e = (1 \times e_0) + (2 \times e_1) + (4 \times e_2) + (8 \times e_3) + (16 \times e_4)$

The mathematical formula for the TransferAmount t conversion is as follows:

$t = 10^e \times m$, for $e = 0$

or

$$t = (10^e \times m) + \sum_{n=1}^{e}\left(2^{14} \times 10^{(n-1)}\right) \text{, for } e > 0$$

where:

t is the TransferAmount;

e is the base 10 exponent;

m is the mantissa; and

n is an integer in the range 1 to e inclusive.

The sign of TransferAmount t is obtained from the value of s given in Table 29 where:

t is positive for s = 0;

t is negative for s = 1.

All TransferAmount conversions shall be rounded up towards positive infinity in favour of the customer (see Table 24 for examples of rounding negative values).

The maximum error due to rounding is 0,055 %. Examples of TransferAmounts and associated errors due to rounding up are shown in Table 25.

Table 24 – Examples of rounding of negative and positive values

Original units to transfer (units of 10^{-5} base currency)	Rounded units transferred (units of 10^{-5} base currency)
–0,99	0
–12,35	–12
–1000,78	–1000
–2314,99	–2314
0,09	1
1000,23	1001
2315,14	2316

DRAFT

Table 25 – Examples of TransferAmounts and rounding errors

Item	Purchase amount (10^{-5} base currency)	e	m	Transfer amount (10^{-5} base currency)	Difference	Rounding error
1	2	0	2	2	0	0,000 %
2	16383	0	16383	16383	0	0,000 %
3	16384	1	0	16384	0	0,000 %
4	16385	1	1	16394	9	0,055 %
5	16386	1	1	16394	8	0,049 %
6	16394	1	1	16394	0	0,000 %
7	16395	1	2	16404	9	0,055 %
8	16404	1	2	16404	0	0,000 %
9	16405	1	3	16414	9	0,055 %
10	180214	1	16383	180214	0	0,000 %
11	180215	2	0	180224	9	0,005 %
12	180216	2	0	180224	8	0,004 %
13	1818524	2	16383	1818524	0	0,000 %
14	1818525	3	0	1818624	99	0,005 %

6.3.7 CRC: CyclicRedundancyCheck

The CRC is a checksum field used to verify the integrity of the data transferred for all tokens, except for Class 0 with SubClass 4 to 7, which uses CRC_C (see 6.3.22). The checksum is derived using the following CRC generator polynomial:

$$x^{16} + x^{15} + x^{2} + 1$$

The total length of the data transferred via the token is 66 bits. The last 16 bits comprise the CRC checksum that is derived from the preceding 50 bits. These 50 bits are left padded with 6 binary zeros to make 56 bits. Before calculation, the CRC checksum is initialised to FFFF hex (see example in Table 26).

Table 26 – Example of a CRC calculation

Original 50 bits	0 00 4A 2D 90 0F F2 hex
Left padded to make 7 bytes	00 00 4A 2D 90 0F F2 hex
Checksum calculated	0F FA hex

DRAFT

6.3.8 Control: InitiateMeterTest/DisplayControlField

The initiate payment meter test data field is 36/28 bits long and is used to indicate the type of test to be performed. The particular test is selected by setting the relevant bit to a logic ONE. The permissible field values are defined in Table 27.

Table 27 – Permissible control field values

Bit No.	Test No	Action	Condition
All bits = 1	0	Do test No. 2 to 5 plus, optionally, any other; inclusion of test No. 2 is mandatory if implemented	Mandatory
1	1	Test supported load switch(es)	Optional
2	2	Test supported display(s) and/or device(s)	Optional
3	3	Display cumulative usage register totals	Mandatory
4	4	Display the KRN and KT value	Mandatory
5	5	Display the TI value	Mandatory
6	6	Test the token input device	Optional
7	7	Display maximum power limit	Optional
8	8	Display tamper status	Optional
9	9	Display active load power	Optional
10	10	Display software version	Mandatory
11	11	Display phase power unbalance limit	Optional
12	12	Display water meter factor (reserved for future definition by the STS Association)	Reserved
13	13	Display tariff rate (reserved for future definition by the STS Association)	Reserved
14	14	Display the EA value	Mandatory
15	15	Display number of key change tokens supported	Mandatory
16	16	Display the SGC value	Mandatory for 3 or 4 KCT meters
17	17	Display the KEN value	Mandatory
18	18	Display the DRN value	Mandatory
19-28/36	Reserved	Reserved for future assignment by the STS Association	Reserved
NOTE In the context of electricity metering the term "usage" refers to either active energy, reactive energy or apparent energy cumulative totals, depending on the specific metering application. In the context of water, gas or time, the meaning may be interpreted in the context of the particular metering application.			

All payment meters shall support test number 0; if any of the incorporated tests are not supported the payment meter shall perform the subset of tests that are supported. The optional selection of additional incorporated tests is subject to the supply agreement between the supplier and the utility and shall then form a normative part of this document.

In the case where a test is optional, the inclusion of this test shall be subject to the supply agreement between the supplier and the utility and shall then form a normative part of this document.

In the case where more than one test is specified on a single token, the behaviour of the payment meter shall be agreed between the utility and the supplier and shall then form a normative part of this document.

DRAFT

6.3.9 MPL: MaximumPowerLimit

The maximum power limit field is a 16-bit field that indicates the maximum power that the load may draw, in watts. Calculation of this field is identical to that of the TransferAmount field (see 6.3.6). See also note in 8.6 for functional requirements of the MeterApplication Process.

6.3.10 MPPUL: MaximumPhasePowerUnbalanceLimit

The maximum phase power unbalance limit field is a 16-bit field that indicates the maximum allowable power difference between phase loads, in watts. Calculation of this field is identical to that of the TransferAmount field (see 6.3.6).

6.3.11 Rate: TariffRate

Reserved for future definition by the STS Association.

6.3.12 WMFactor: WaterMeterFactor

Reserved by the STS Association for water application.

6.3.13 Register: RegisterToClear

A unique 16-bit binary value in the range 0 to FFFF hex; to select the particular register that should be cleared with the ClearCredit token. The defined values are given in Table 28.

Table 28 – Selection of register to clear

Value	Action
0	Clear Electricity Credit register
1	Clear Water Credit register
2	Clear Gas Credit register
3	Clear Time Credit register
4	Clear Electricity Currency Credit register
5	Clear Water Currency Credit register
6	Clear Gas Currency Credit register
7	Clear Time Currency Credit register
8 to FFFE hex	Reserved for future assignment by the STS Association
FFFF hex	Clear all Credit registers in the payment meter

6.3.14 NKHO: NewKeyHighOrder

The high order 32 bits of the new DecoderKey that has been generated (see 6.4.4) and which is to be transferred to the payment meter by means of the token.

6.3.15 NKLO: NewKeyLowOrder

The low order 32 bits of the new DecoderKey that has been generated (see 6.4.4) and which is to be transferred to the payment meter by means of the token.

6.3.16 NKMO1: NewKeyMiddleOrder1

The second most significant 32 bits of the 128-bit DecoderKey that has been generated (see 6.4.4) and which is to be transferred to the payment meter by means of a token.

6.3.17 NKMO2: NewKeyMiddleOrder2

The third most significant 32 bits of the 128-bit DecoderKey that has been generated (see 6.4.4) and which is to be transferred to the payment meter by means of a token.

6.3.18 KENHO: KeyExpiryNumberHighOrder

This is the high order 4 bits of the KEN (see 6.1.10).

6.3.19 KENLO: KeyExpiryNumberLowOrder

This is the low order 4 bits of the KEN (see 6.1.10).

6.3.20 RO: RolloverKeyChange

The RO bit shall be set to 1 in the Set1stSectionDecoderKey token when the BaseDate associated with the destination VendingKey/DecoderKey is later than the BaseDate associated with the source VendingKey/DecoderKey and shall be set to 0 otherwise.

If the RolloverKeyChange bit is set = 1, the payment meter shall perform a roll over key change. This operation is identical to a normal key change, except that the TID memory store in the payment meter is filled with token identifiers of value 0 (zero).

6.3.21 S&E: SignAndExponent

The bit positions for extraction of S&E variables s, e_4, e_3 and e_2 are given in Table 29. For the assignment of values to s and e, see 6.3.6.3.

Table 29 – S&E bit positions for variables s, e_4, e_3 and e_2

Bit position	3	2	1	0
Variable	s	e_4	e_3	e_2

6.3.22 CRC_C: CyclicRedundancyCheck_C

The CRC_C is a checksum field used to verify the integrity of the data transferred for token Class 0 with SubClass 4 to 7 and is calculated as defined in 6.3.7, but with the following change:

A single byte with the value of 01 hex is appended to the 56-bit value before calculation starts. An example of a CRC_C calculation is given in Table 30.

Table 30 – Example of a CRC_C calculation

Original 50 bits	0 00 4A 2D 90 0F F2 hex
Left padded to make 7 bytes	00 00 4A 2D 90 0F F2 hex
01 hex appended to the end	00 00 4A 2D 90 0F F2 01 hex
Checksum calculated	7BC4 hex

6.4 TCDUGeneration functions

6.4.1 Definition of the TCDU

The TCDU may be different for each TokenCarrierType and is therefore defined separately for each physical layer protocol standard relevant to each part of the IEC 62055-5x series.

DRAFT

6.4.2 Transposition of the Class bits

This function is used by other TCDUGeneration functions (see 6.4.3 to 6.4.5). It inserts the 2 Class bits into the 64-bit data stream to make a 66-bit number according to the method outlined below.

The 64-bit number has its least significant bit in bit position 0 and its most significant bit in bit position 63. The 64-bit binary number string is modified to include the unencrypted token Class The 2-bit token Class value is inserted to occupy bit positions 28 and 27. The original values of bit positions 28 and 27 are relocated to bit positions 65 and 64. The most significant bit of the token Class now occupies bit position 28. The process is shown in Figure 6.

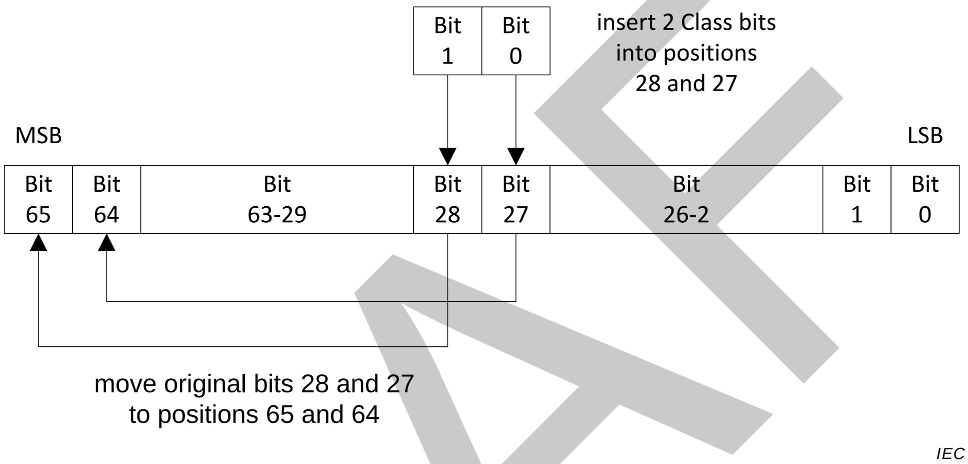


Figure 6 – Transposition of the 2 Class bits

Example: Insertion of the token Class = 01 (binary).

The 64-bit binary number grouped in nibbles (Bits 27 and 28 highlighted in bold):

0110 0101 0100 0011 0010 0001 0000 1001 100**0** **0**111 0110 0101 0100 0011 0010 0001

Copy bits 28 and 27 into bit positions 65 and 64, creating a 66-bit number:

00 0110 0101 0100 0011 0010 0001 0000 1001 1000 0111 0110 0101 0100 0011 0010 0001

Replace bits 28 and 27 with the 2 Class bits:

00 0110 0101 0100 0011 0010 0001 0000 1001 100**0** **1**111 0110 0101 0100 0011 0010 0001

6.4.3 TCDUGeneration function for Class 0,1 and 2 tokens

This is the transfer function from the APDU to the TCDU (see Figure 7) and is applicable to all Class 0, Class 1 and Class 2 tokens, except for the key change tokens (see 6.2.7 and 6.2.8).

NOTE 1 The data elements in the APDU are defined in 6.1.1.

NOTE 2 The data elements in the TCDU are defined in part of the IEC 62055-5x series physical layer protocol standard relevant to the specific TCT of interest.

DRAFT

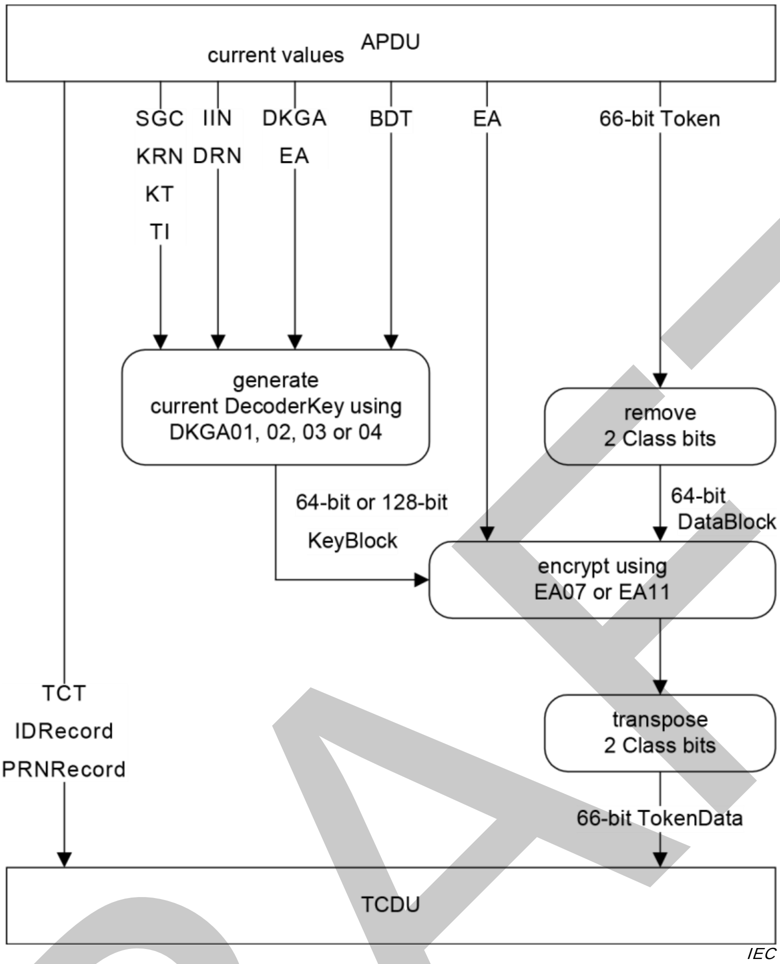


Figure 7 – TCDUGeneration function for Class 0, 1 and 2 tokens

The transfer function for Class 0 and Class 2 tokens is outlined as follows:

- The 2 Class bits are removed from the 66-bit token to yield a 64-bit result, which is then presented to the encryption algorithm as its DataBlock input. The specific algorithm to use is in accordance with the EA code in the APDU;
- The KeyBlock input for the encryption algorithm is obtained from the decoder key generation algorithm, which generates the current DecoderKey using the current values of SGC, KRN, KT, TI, IIN, DRN, DKGA, EA and BDT from the APDU as indicated. The specific decoder key generation algorithm to use is in accordance with the value of DKGA in the APDU;
- After encryption the 2 Class bits are again re-inserted into the 64-bit number in accordance with the method defined in 6.4.2 to yield a 66-bit result, which is populated into the TokenData field of the TCDU in accordance with the particular definition in the relevant physical layer protocol standard;
- Similarly the TCT, IDRecord and PRNRecord data elements from the APDU are transferred to the TCDU as indicated, into the appropriate fields of the TCDU in accordance with the particular definition in the relevant physical layer protocol standard.

The transfer function for Class 1 tokens is identical to the TCDUGeneration function for Class 0 and Class 2 tokens, except that the token does not get encrypted. The function is outlined as follows:

- The 2 Class bits are removed from the 66-bit token and transposed in accordance with the method defined in 6.4.2 to yield a 66-bit result, which is populated into the TokenData field of the TCDU in accordance with the particular definition in the relevant physical layer protocol standard;

- Similarly the TCT, IDRecord and PRNRecord data elements from the APDU are transferred to the TCDU as indicated, into the appropriate fields of the TCDU in accordance with the particular definition in the relevant physical layer protocol standard.

6.4.4 TCDUGeneration function for key change tokens

This is the transfer function from the APDU to the TCDU (see Figure 8) and is applicable to all key change tokens.

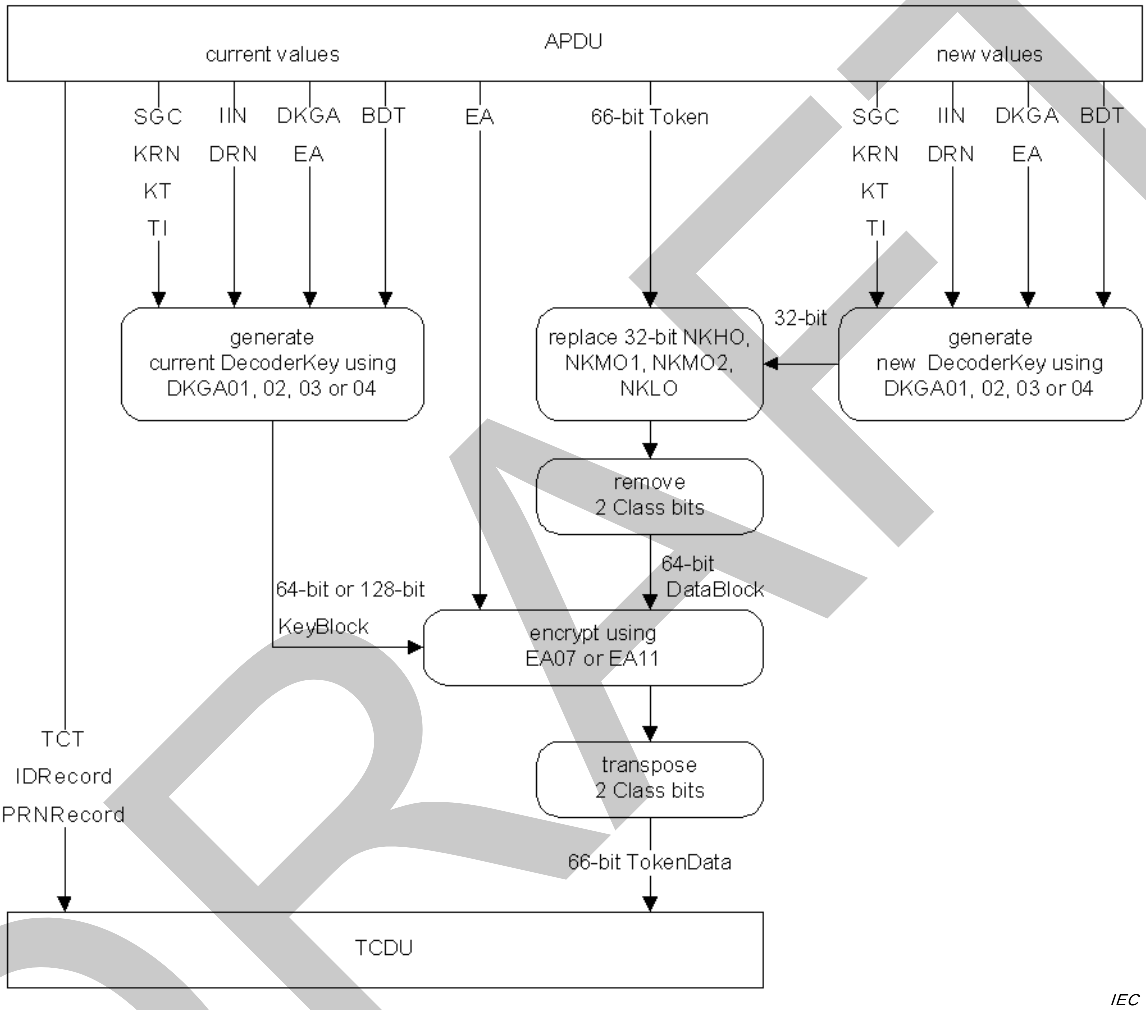


Figure 8 – TCDUGeneration function for key change tokens

A separate TCDU is produced for each key change token in the set.

Note that the APDU has to present two sets of data for the PANBlock and CONTROLBlock: one set with the new data for the new DecoderKey and a second set with the current data for the current DecoderKey. The DKGA value is the same for both sets.

NOTE 1 The data elements in the APDU are defined in 6.1.1.

NOTE 2 The data elements in the TCDU are defined in each part of the IEC 62055-5x series physical layer protocol standard relevant to the specific TCT of interest.

The transfer function is outlined as follows:

- the new DecoderKey is generated using the new values of SGC, KRN, KT, TI, IIN, DRN, DKGA, EA and BDT. The specific algorithm to use is in accordance with the value of DKGA in the APDU;

DRAFT

- the resultant new DecoderKey value 32-bit portion is then used to replace the NKHO, NKMO1, NKMO2 or NKLO field of the key change token (see 6.2.7 and 6.2.8) as presented by the APDU;
- the 2 Class bits are removed from the 66-bit token to yield a 64-bit result, which is then presented to the encryption algorithm as its DataBlock input. The specific encryption algorithm to use is in accordance with the EA code in the APDU;
- the KeyBlock input for the encryption algorithm is obtained from the decoder key generation algorithm, which generates the current DecoderKey using the current values of SGC, KRN, KT, TI, IIN, DRN DKGA, EA and BDT from the APDU as indicated. The specific decoder key generation algorithm to use is in accordance with the value of DKGA in the APDU;
- after encryption, the 2 Class bits are again re-inserted into the 64-bit number in accordance with the method defined in 6.4.2 to yield a 66-bit result, which is populated into the TokenData field of the TCDU in accordance with the particular definition in the relevant physical layer protocol standard;
- similarly the TCT, IDRecord and PRNRecord data elements from the APDU are transferred to the TCDU as indicated, into the appropriate fields of the TCDU in accordance with the particular definition in the relevant physical layer protocol standard.

6.4.5 TCDUGeneration function for Set2ndSectionDecoderKey token

This is now incorporated into 6.4.4.

6.5 Security functions

6.5.1 General requirements

With the exception of DITK values, VendingKey and DecoderKey values shall only be generated by a device responsible for token generation, such as a POS that is certified as STS-compliant and which is subject to an STS-certified KeyManagementSystem (see Clause 9). This subclause describes the key generation methods used by such devices and is applicable to manufacturers of these devices.

6.5.2 Key attributes and key changes

6.5.2.1 Key change requirements

With the exception of DITK values, STS key values shall only be introduced or changed in a payment meter from a device responsible for key management, such as a POS that is certified as STS-compliant, and which is subject to STS key management. This subclause describes the STS key change method used between such devices and payment meters, and is applicable to manufacturers of these devices and payment meters.

An STS key change provides the mechanism for changing the DecoderKey present in a decoder from its current value to a new value. This process may be initiated by several events or circumstances, including the following:

- a new or repaired payment meter that contains a manufacturer's DITK value shall be changed before leaving the manufacturing or repair premises to contain the appropriate value of manufacturer's default (DDTK) or utility's DecoderKey (DUTK or DCTK) depending on the SupplyGroup to which the payment meter has been allocated;
- a SupplyGroup's VendingKey has either expired or been compromised, and is replaced by a new VendingKey revision and, as a result, each DecoderKey within the SupplyGroup shall be changed from its current DecoderKey value to the DecoderKey value that corresponds to the new VendingKey value;
- a payment meter is re-allocated from one SupplyGroup to another SupplyGroup and, as a result, its DecoderKey shall be changed from its current value generated from the previous SupplyGroup VendingKey to the new value generated from its new SupplyGroup VendingKey; or

- the TI for a payment meter is changed and, as a result, its DecoderKey shall be changed from its current value (that corresponds to the previous TI) to the new value (that corresponds to the new TI).

The key change token set effects an STS key change. This meter-specific management token set transfers the following information from the POS to the payment meter, encrypted under the current DecoderKey:

- the value of the new DecoderKey;
- the KEN;
- the KRN;
- the KT;
- the SGC (only in the case of the three-token set and the four-token set);
- the TI.

An STS key change process for a payment meter shall be initiated whenever any one of the following attributes of the VendingKey changes in value:

- the value of the VendingKey;
- the value of BDT;
- the value of the SGC;
- the value of the TI;
- the value of the KEN;
- the value of the KRN;
- the value of the KT;
- the value of the DKGA.

NOTE See 6.1.1 for detailed specifications on the data elements in the APDU and 6.5.3 for DKGA requirements.

A particular SGC may be associated with more than one VendingKey at the same time during its operational life, in which case each VendingKey shall be identified by its associated KRN.

Key change tokens shall not be generated in the case where the destination key's KEN relative to BDT is in the past (according to the system clock).

Key change tokens shall not be generated where the BaseDate associated with the destination VendingKey/DecoderKey is earlier than the BaseDate associated with the source VendingKey/DecoderKey.

A POS may optionally generate and issue key change tokens automatically or manually, but this shall be specified in the purchase agreement between the manufacturer and the utility.

6.5.2.2 VendingKey classification

6.5.2.2.1 Classification of vending keys

The VendingKey is a cryptographic key value that is secretly generated, stored and distributed within the KeyManagementSystem (see Annex A). VendingKeys are the seed keys from which DecoderKeys are generated.

The VendingKey is classified according to its associated KT value, which is an attribute that defines the purpose for which the key can be used. Three KT values are defined for VendingKeys and correspond to three of the SupplyGroup types (see 6.1.6), namely Default, Unique and Common. The VendingKey for a given SupplyGroup is the seed key used to generate the DecoderKey values for all payment meters within the SupplyGroup.

DRAFT

STS VendingKeys are classified according to the KT values given in Table 31.

Table 31 – Classification of vending keys

KT	SGC type	VendingKey type	Context
0	Initialization	Not specified	Not applicable
1	Default	VDDK	VendingDefaultDerivationKey
2	Unique	VUDK	VendingUniqueDerivationKey
3	Common	VCDK	VendingCommonDerivationKey

At any given moment, a unique VDDK value exists for each Default SupplyGroup defined. Similarly, a unique VUDK value for each Unique SupplyGroup and a unique VCDK value for each Common SupplyGroup are defined.

6.5.2.2.2 VDDK: VendingDefaultDerivationKey

This type of key is used as the seed key for generation of DDTK values – it shall not be used to generate DITK, DUTK or DCTK values.

6.5.2.2.3 VUDK: VendingUniqueDerivationKey

This type of key is used as the seed key for generation of DUTK values – it shall not be used to generate DITK, DDTK or DCTK values.

6.5.2.2.4 VCDK: VendingCommonDerivationKey

This type of key is used as the seed key for generation of DCTK values – it shall not be used to generate DITK, DDTK or DUTK values.

6.5.2.3 DecoderKey classification

6.5.2.3.1 Classification of decoder keys

STS DecoderKeys are classified according to the KT values given in Table 32 and inherit their type from that of the VendingKey, from which they are derived.

Table 32 – Classification of decoder keys

KT	SGC type	DecoderKey type	Context
0	Initialization	DITK	DecoderInitialisationTransferKey
1	Default	DDTK	DecoderDefaultTransferKey
2	Unique	DUTK	DecoderUniqueTransferKey
3	Common	DCTK	DecoderCommonTransferKey

For further information regarding the rules for changing of a key from one type to another type, see Figure 9 and Table 33 in 6.5.2.4.

A payment meter shall be capable of storing at least one DecoderKey value and its associated KT value in its DecoderKeyRegister (see 7.3.2).

It shall not be possible for the DecoderKey value to be read or retrieved from a payment meter under any circumstances, whether encrypted or in the clear.

DRAFT

6.5.2.3.2 DITK: DecoderInitialisationTransferKey

DITK values are used to initialise the DecoderKeyRegister during production or repair at the manufacturer's premises. These keys are the property of the MeterManufacturer. As such, they are generated and managed by the manufacturer, and are unknown to the utility.

No payment meter purchased by the utility shall leave a manufacturer's premises with a DITK value in the DecoderKeyRegister. The DecoderKeyRegister shall contain either a DDTK, DUTK or DCTK value supplied by the KMC. A DITK is the only key type that can be introduced into a payment meter as a plaintext value. DDTK, DUTK or DCTK values can only be introduced into a payment meter as cipher text (encrypted) values.

A DITK shall only be used for the following key management functions:

- as the parent key for another DITK; in other words, to encrypt another DITK for the purpose of introducing it into the DecoderKeyRegister;
- as the parent key for a DDTK;
- as the parent key for a DUTK, and
- as the parent key for a DCTK, but only in a payment meter using an erasable magnetic card as a token carrier (for TCT value = 01).

The above functions may be performed via the key change token set or via a manufacturer proprietary loading mechanism that utilizes the key change token set. The payment meter should only accept the DDTK, DUTK or DCTK encrypted under the DITK supplied by the manufacturer in the key change token set format.

It is the responsibility of the manufacturer to ensure that appropriate security measures are applied to any DITK so that DDTK, DUTK or DCTK values encrypted with a DITK cannot be compromised.

A DITK can also be used to decrypt other meter-specific management functions. It can be used to decrypt an STS credit transfer function; in other words, a valid STS TransferCredit token can be decrypted and applied by a payment meter that contains a DITK in its key register in order to facilitate testing of the payment meter during production or repair.

6.5.2.3.3 DDTK: DecoderDefaultTransferKey

DDTK values are used to support payment meters allocated to a default SupplyGroup. A payment meter that has not been allocated to a Common SupplyGroup or a Unique SupplyGroup at the time of manufacture or repair cannot be loaded with its corresponding DCTK or DUTK value. Instead it is allocated to a Default group unique to each manufacturer and loaded with its corresponding DDTK value. Each MeterManufacturer receives a unique VDDK, from which he generates all DDTK values for installation into payment meters during manufacture.

Subsequently, at the time of installation or operation, a payment meter that has now been re-allocated to another specific SupplyGroup can be loaded with the corresponding DUTK or DCTK value, encrypted under its parent DDTK. DDTK values are the property of the respective MeterManufacturer or Utility and are managed within the KeyManagementSystem.

A DDTK is a secret value, and shall not be accepted by a payment meter as a plaintext value. A payment meter shall only load a DDTK if it is encrypted under the parent DecoderKey present in the DecoderKeyRegister.

A DDTK shall only be used for the following key management functions:

- as the parent key for another DDTK; in other words, to encrypt another DDTK for the purpose of introducing it into the DecoderKeyRegister;

DRAFT

- as the parent key for a DUTK, and
- as the parent key for a DCTK, but only in a payment meter using an erasable magnetic card as a token carrier (for TCT value = 01).

The above functions may be performed via the key change token set, or via a manufacturer's proprietary loading mechanism that utilizes the key change token set. A DDTK shall not be used to decrypt a DITK for the purpose of introducing it into the DecoderKeyRegister.

A DDTK can also be used to decrypt other meter-specific management functions. It shall not be used to decrypt and accept an STS credit transfer function; in other words, a valid TransferCredit token shall not be accepted by a payment meter that contains a DDTK in its DKR, even if the TransferCredit token has been encrypted with the same DDTK value.

NOTE The emphasis is on the acceptance and not on the decryption of the TransferCredit token.

Similarly a POS device used for encrypting tokens shall not encrypt TransferCredit tokens using DDTK values (see also 6.5.2.4).

6.5.2.3.4 DUTK: DecoderUniqueTransferKey

DUTK values are used to support payment meters allocated to a unique SupplyGroup. A payment meter that has been allocated to a unique SupplyGroup at the time of manufacture or repair can be loaded with its DUTK value that corresponds to the unique group and that has been encrypted under a parent DITK. Subsequently, at the time of installation or operation, a payment meter, which has to be re-allocated to another unique group can be loaded with the corresponding DUTK value, encrypted under a parent DUTK.

A DUTK is a secret value, and shall not be accepted by a payment meter as a plaintext value. A payment meter shall only load a DUTK if it has been encrypted under the parent DecoderKey present in the DecoderKeyRegister. DUTK values are the property of the respective utility and are managed within the KeyManagementSystem.

A purchased or repaired payment meter that leaves the manufacturer's premises may contain a DUTK value supplied by the KMC in the DecoderKeyRegister.

A DUTK shall only be used for the following key management functions:

- as the parent key for another DUTK; in other words, to encrypt another DUTK for the purpose of introducing it into the DecoderKeyRegister; and
- as the parent key for a DDTK.

The above functions may be performed via the key change token set, or via a manufacturer's proprietary loading mechanism that utilizes the key change token set. A DUTK shall not be used to decrypt a DITK or a DCTK for the purpose of loading it into the DecoderKeyRegister. Similarly a DUTK shall not be used to encrypt a DITK or a DCTK for the purpose of transferring it to the payment meter in the form of a token.

A DUTK can also be used to encrypt or decrypt other meter-specific management functions. It can be used to encrypt or decrypt a STS credit transfer function; in other words, a valid TransferCredit token can be encrypted or decrypted and applied by a payment meter that contains a DUTK in its DKR.

6.5.2.3.5 DCTK: DecoderCommonTransferKey

DCTK values are used to support payment meters that use erasable magnetic card token carriers (i.e. TCT value = 01) and that are allocated to common SupplyGroups. A payment meter that has been allocated to a common SupplyGroup at the time of manufacture or repair can be loaded with the DCTK value that corresponds to the common SupplyGroup and that has been encrypted under a parent DITK. Subsequently, at the time of installation or operation, a payment

DRAFT

meter that has to be re-allocated to another common SupplyGroup can be loaded with the corresponding DCTK value that has been encrypted under a parent DCTK.

A DCTK shall only be used with payment meters that use erasable magnetic card token carriers (TCT value = 01) and shall only be accepted by such payment meters. Payment meters with any other token carrier types (TCT value > 01) shall reject tokens encrypted under DCTK values.

POS encryption devices shall not encrypt tokens using DCTK values other than for erasable magnetic card token carriers (TCT value = 01).

A DCTK is a secret value, and shall not be accepted by a payment meter as a plaintext value. A payment meter shall only load a DCTK if it has been encrypted under the parent DecoderKey present in the DecoderKeyRegister. DCTK values are the property of the respective utility and are managed within the KeyManagementSystem.

A purchased or repaired payment meter with an erasable magnetic card token carrier (TCT value = 01) that leaves the manufacturer's premises may contain a DCTK value supplied by the KMC in the DecoderKeyRegister.

A DCTK shall only be used for the following key management functions:

- as the parent key for another DCTK; in other words, to encrypt another DCTK for the purpose of introducing it into the DecoderKeyRegister;
- as the parent key for a DDTK; and
- as the parent key for a DUTK.

The above functions may be performed via the key change token set, or via a manufacturer’s proprietary loading mechanism that utilizes the key change token set. A DCTK shall not be used to decrypt a DITK for the purpose of introducing it into the DecoderKeyRegister. Similarly a DCTK shall not be used to encrypt a DITK for the purpose of transferring it to the payment meter in the form of a token.

A DCTK can also be used to encrypt or decrypt other meter-specific management functions. It can be used to encrypt or decrypt a STS credit transfer function; in other words, a valid TransferCredit token can be encrypted or decrypted and applied by a payment meter that contains a DCTK in its DKR and that uses a magnetic card token carrier (TCT value = 01).

6.5.2.4 State diagram for DecoderKey changes

Figure 9 illustrates the KT states that a DecoderKey may assume from time to time.

DRAFT

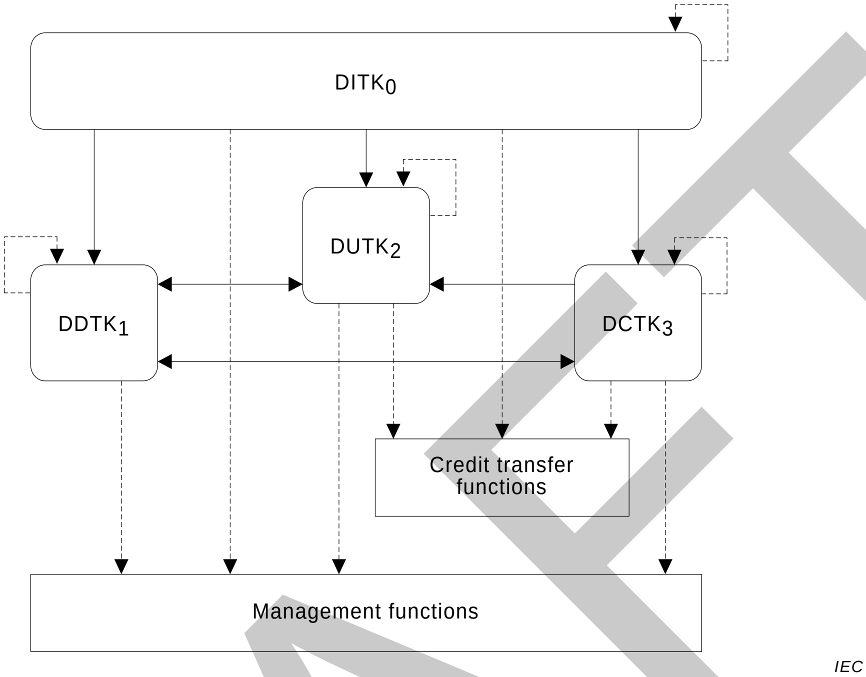


Figure 9 – DecoderKey changes – state diagram

Where one key is used to encrypt another key (as in the key change token set), the former is referred to as the parent key and the latter as the child key.

The solid line arrows indicate the direction in which a key may change from one type to another type. The type that it changes from is the parent key and the type that it changes to is the child key. To effect a change of the DecoderKey the new key (or child key) is encrypted with the parent key and then loaded into the payment meter by means of a key change token set. The payment meter then replaces the parent key with the child key, which now becomes the new parent key.

The dotted line arrows indicate the function, for which a KT may be used, i.e. the values that it may encrypt or decrypt. For example, only a DITK, DUTK or DCTK can be used to encrypt or decrypt a credit transfer function, but all four types can be used to encrypt or decrypt meter-specific management functions.

Table 33 details the permitted key change state relationships and associated functions.

The child key rows refer to the permitted usage of decoder key types for encryption of DecoderKeys in the key change token set key management functions. Similarly, the management and credit rows detail the permitted usage of decoder key types for the encryption of the remaining meter-specific management functions and credit transfer functions respectively.

Table 33 – Permitted relationships between decoder key types

Child key	Permitted usage			
	Parent key			
	$DITK_0$	$DDTK_1$	$DUTK_2$	$DCTK_3$
$DITK_0$	Yes	No	No	No
$DDTK_1$	Yes	Yes	Yes	Yes[a]
$DUTK_2$	Yes	Yes	Yes	Yes[a]
$DCTK_3$	Yes[a]	Yes[a]	No	Yes[a]
Management function	Yes	Yes	Yes	Yes[a]
Credit function	Yes	No	Yes	Yes[a]
[a] For payment meters with TCT = 01 only.				

The key type relationship policy in the POS shall be enforced in a secure device such as a tamper-proof CryptographicModule.

6.5.2.5 KeyRevisionNumber (KRN)

Each SupplyGroup has one or more VendingKeys associated with it. A KRN uniquely identifies a VendingKey within the SupplyGroup. Together the SGC and KRN uniquely identify a VendingKey.

The KRN is a single decimal digit with a range of 1, 2, .. 9. The association between SGC, KRN, and VendingKey is set by the KMS. The first VendingKey for a SupplyGroup should be assigned KRN 1; successive VendingKeys are assigned successive revision numbers until KRN 9 at which state the sequence begins again at 1.

At any given moment there may be no more than 9 successive VendingKey revisions present in a POS for a given SupplyGroup.

A payment meter's DecoderKey is associated with the SGC and KRN of the VendingKey from which it is derived. A payment meter is required to store the KRN associated with the DecoderKey, as passed in the key change token set (see also 7.3.2).

The concept of key revision only applies to VDDK, VUDK and VCDK VendingKey types and DDTK, DUTK and DCTK DecoderKey types. A DITK shall not be associated with a KRN.

All payment meters within a SupplyGroup should be set to the latest VendingKey for that SupplyGroup. This information is managed by the management system and if for any reason the KRN in the payment meter is not the same as the KRN of the latest VendingKey for the SupplyGroup as recorded in the management system, this condition shall be corrected by means of an appropriate change of the DecoderKey (see also 6.5.2.1 and C.13.2.4).

NOTE The KRN does not determine the latest VendingKey for a given SGC. This is managed by means of other control attributes such as active date and expiry date, which are outside the scope of this document. Examples of these may be found in STS 600-4-2, *Standard Transfer Specification – Companion Specification – Key Management System* (see Bibliography).

DRAFT

6.5.2.6 KeyExpiryNumber (KEN)

A KEN is associated with each VendingKey by the KMS, and defines the following:

- the time-period, after which the VendingKey expires, and may no longer be used by a POS to generate DecoderKeys for the purpose of encrypting TransferCredit tokens, or meter-specific management tokens that incorporate the TID field;
- the time-period, after which the VendingKey expires, and may no longer be used by a POS to generate DecoderKeys for the purpose of encoding into a Key Change Token set as the new DecoderKey;
- the time-period, after which any DecoderKey generated from the VendingKey expires, and may no longer be used by a payment meter to accept TransferCredit tokens, or meter-specific management tokens that incorporate the TID field. Implementation of this by a payment meter is optional.

The required value of the KEN shall be transferred to the payment meter in the KENHO and KENLO fields of the key change token set (see 6.2.7 and 6.2.8).

The KEN is an 8-bit number (range 0 – 255) that expresses this period as a displacement relative to the STS base date token identifier time stamp (see 6.3.5.1). Each unit in the KEN corresponds to a period of duration 2^{16}–1 (65535) min, and there are 2^{8} (256) of these periods numbered 0, 1. .255 before the current STS base date time stamp is replaced by the next STS base time stamp. Thus the KEN corresponds to the most significant 8 bits of the 24-bit TID. Any token identifier whose most significant 8 bits are greater than a given key's KEN shall not be encrypted or decrypted with that key.

A POS may not issue a TransferCredit token encrypted under a DecoderKey whose corresponding VendingKey has expired. This is simple to verify by comparing the most significant 8 bits of the TID with the KEN corresponding to the VendingKey; if it is greater, the VendingKey has expired and may no longer be used to generate a DecoderKey to encrypt the TransferCredit token. It also cannot be used to generate a DecoderKey to encrypt any meter-specific management tokens that utilize the TID field. This does not apply to the key change token set that does not utilize the TID field. Hence, an expired DecoderKey can still be used to encrypt its replacement DecoderKey for the purpose of a DecoderKey change.

A payment meter can optionally implement key expiry and store the KEN that corresponds to its current DecoderKey, as passed in the key change token set. All tokens that are entered into the payment meter, and that incorporate a token identifier field, are validated against this KEN. If the most significant 8 bits of the TID are greater than this KEN, the token shall be rejected.

Where implemented, the concept of key expiry only applies to VendingKey values of type VDDK, VUDK and VCDK, and DecoderKey values of type DDTK, DUTK and DCTK that can be generated from the corresponding vending key types. A DITK shall not be associated with a KEN.

The management of the KEN by the KMS shall comply with the relevant Code of practice.

See also C.3.4 for Code of practice on managing this data element.

6.5.3 DecoderKey generation

6.5.3.1 PANBlock construction

The 16 digit PANBlock is constructed from data elements extracted from the MeterPAN in the APDU as defined in Table 34 and Table 35.

The most significant digit is in position 15 and the least significant digit in position 0.

DRAFT

Table 34 – Definition of the PANBlock

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	I	I	I	I/D	I/D	D	D	D	D	D	D	D	D	D	D	D

Table 35 – Data elements in the PANBlock

Digit	Name	Format	Reference
I	IIN	Range 0 to 9 hex per digit	6.1.2.2
D	DRN	Range 0 to 9 hex per digit	6.1.2.3

For DDTK and DUTK coded decoders, the following applies:

- Where the DRN is 11 digits long, the PANBlock is made up of the 5 least significant digits of the IIN and the 11 digits of the DRN. The 11 digits of the DRN take up positions 10 to 0 in the PANBlock and the 5 least significant digits of the IIN take up positions 15 to 11 in the PANBlock;
- Where the DRN is 13 digits long, the PANBlock is made up of the 3 least significant digits of the IIN and the 13 digits of the DRN. The 13 digits of the DRN take up positions 12 to 0 in the PANBlock and the 3 least significant digits of the IIN take up positions 15 to 13 in the PANBlock;

If the IIN is of insufficient length to make up the 16 digits, the digits extracted are right justified within the block and padded on the left with zeroes (for example, for an IIN of 600727 and a DRN of 12345678903, the PANBlock is 0072712345678903).

For a DDTK or DUTK the actual designated DRN is used, but for a DCTK the DRN digits are set to zeros in the PANBlock, thus it always uses a fixed value of 0072700000000000.

6.5.3.2 CONTROLBlock construction

The 16 digit CONTROLBlock is constructed from the data elements in the APDU as defined in Table 36 and Table 37.

The most significant digit is in position 15 and the least significant digit in position 0.

Table 36 – Definition of the CONTROLBlock

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	C	S	S	S	S	S	S	T	T	R	F	F	F	F	F	F

Table 37 – Data elements in the CONTROLBlock

Digit	Name	Format	Reference
C	KT digit	Range 0 to 3 hex per digit, 4 to F hex = reserved for future assignment by the STS Association	6.1.9
S	SGC digit	Range 0 to 9 hex per digit	6.1.6
T	TariffIndex digit	Range 0 to 9 hex per digit	6.1.7
R	KRN digit	Range 1 to 9 hex per digit	6.1.8
F	Pad value digit	Always F hex per digit	x

6.5.3.3 DKGA01: DecoderKeyGenerationAlgorithm01

This DecoderKeyGenerationAlgorithm01 is to be used on a small limited set of defined DRN values only. It is included in this document to maintain backward compatibility with a limited number of legacy STS-compliant payment meters of an early generation also using the STA (EA code 07). The POSApplicationProcess gives the appropriate directive by means of the DKGA code in the APDU.

The DecoderKey is diversified from a 64-bit single DES VendingKey value.

This DecoderKeyGenerationAlgorithm01 is applicable to all payment meters that meet all of the following criteria:

- using IIN = 600727;
- and the KRN = 1;
- and the KT = 1 or 2 (default or unique);
- and the EA code 07 (STA)
- and the DRN falls within the ranges listed in Table 38.

Table 38 – Range of applicable decoder reference numbers

Decoder reference numbers		
0109000000X	to	0109000499X
0100000000X	to	0100499999X
0300000000X	to	0311400000X
0400000000X	to	0405999999X
0601000000X	to	0603999999X
0640000000X	to	0641999999X
0666000000X	to	0669999999X
0699000001X	to	0699000999X
0700000000X	to	0702099999X
NOTE X is a check digit, the value of which varies in accordance with the value of the preceding 10 digits (see 6.1.2.3).		

This DecoderKeyGenerationAlgorithm01 is also applicable to all payment meters that meet all of the following criteria:

- using IIN = 600727;
- and the KRN = 1;
- and the KT = 3 (common);
- and the EA code 07 (STA);
- and coded with one of the SGC values listed in Table 39.

Table 39 – List of applicable supply group codes

Supply group code
100702
990400
990401
990402
990403
990404
990405

The process flow for the DKGA01 is shown in Figure 10.

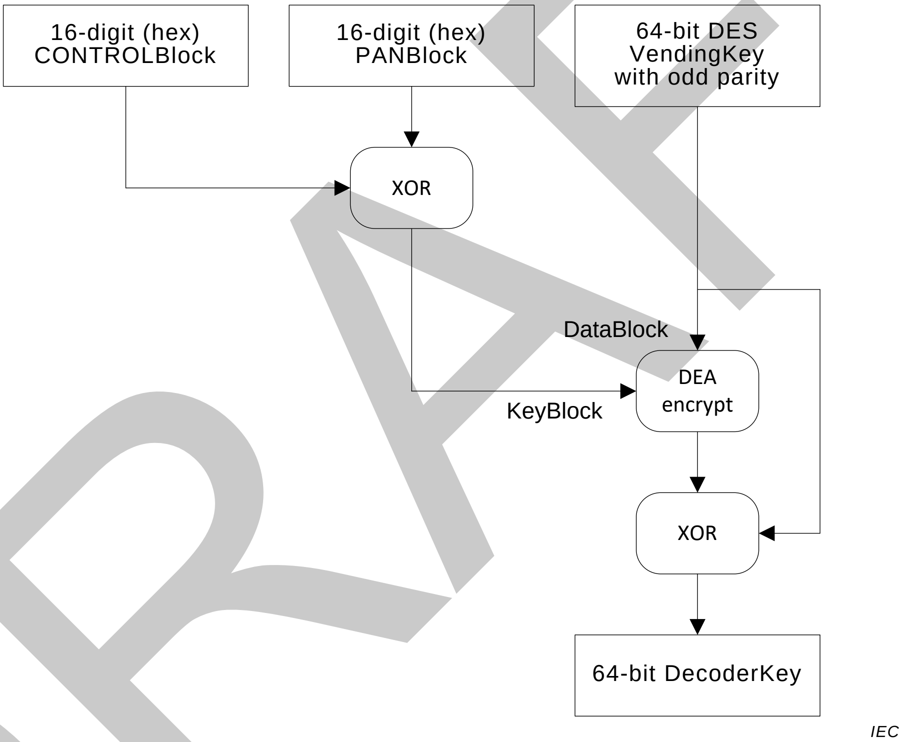


Figure 10 – DecoderKeyGenerationAlgorithm01

Construct the 64-bit PANBlock and the 64-bit CONTROLBlock as defined in 6.5.3.1 and 6.5.3.2.

The encryption algorithm is DEA in accordance with FIPS PUB 46-3, single DES in ECB mode, using a single 64-bit DES VendingKey with odd parity.

In this instance the 64-bit DES VendingKey is used as the conventional DataBlock input to the DEA, while the resultant XOR of the CONTROLBlock with the PANBlock is used as the conventional KeyBlock input to the DEA. In other words, the data and key input blocks are swapped with respect to the conventional configuration.

6.5.3.4 DKGA02: DecoderKeyGenerationAlgorithm02

The DecoderKeyGenerationAlgorithm02 may be used for all payment meters that do not meet the criteria for selecting DecoderKeyGenerationAlgorithm01. The POS ApplicationProcess gives the appropriate directive by means of the DKGA code in the APDU.

The DecoderKey is diversified from a 64-bit single DES VendingKey value.

The process flow for the DKGA02 is shown in Figure 11.

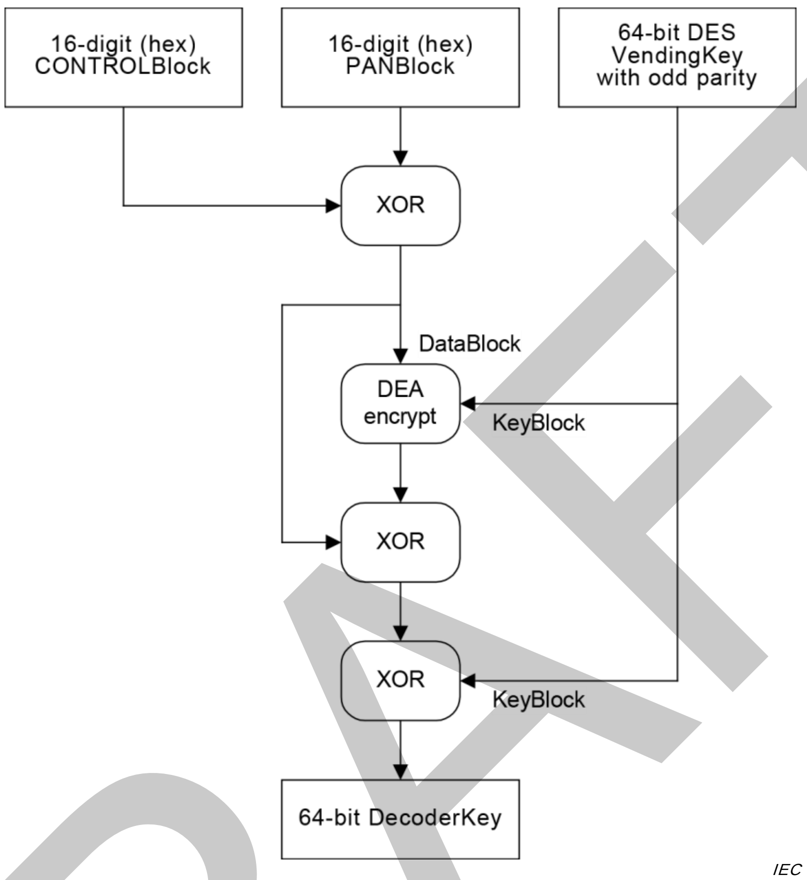


Figure 11 – DecoderKeyGenerationAlgorithm02

Construct the 64-bit PANBlock and the 64-bit CONTROLBlock as defined in 6.5.3.1 and 6.5.3.2.

Encryption is DEA in accordance with FIPS PUB 46-3, single DES in ECB mode, using a single 64-bit DES VendingKey with odd parity.

6.5.3.5 DKGA03: DecoderKeyGenerationAlgorithm03

This algorithm is deprecated and shall not be used for development of new products.

6.5.3.6 DKGA04: DecoderKeyGenerationAlgorithm04

KDF-HMAC-SHA-256 is a NIST SP800-108 Key Derivation Function (KDF) in Feedback mode using no Initialization Vector (IV) and no counter, with HMAC-SHA-256 as the Pseudo-random Function, and with field L a 32-bit binary value with MSB-first.

DKGA04 shall use the KDF-HMAC-SHA-256 algorithm, where HMAC is defined in ISO 9797-2 and SHA-256 is defined in ISO 10118-3. KDF-HMAC-SHA-256 is the HMAC standard applied to SHA-256 standard.

The process flow for the DKGA04 is outlined as follows:

- Construct the 49-byte DataBlock as given in Table 40 with Field No 1 being the left-most position and Field No 17 being the right-most position;

- Present a 160-bit VendingKey to the KDF-HMAC-SHA-256 function;
- Set the DecoderKey key length to 64 bits for EA07 or 128 bits for EA11;
- Calculate the DecoderKey and truncate it to 64 or 128 bits, retaining the left most-significant bits.

Thus DK = Left(HMAC-SHA-256(VK, DataBlock), L), where Left(X, Len) truncates the value X keeping the Len leftmost bits.

It shall not be possible to calculate a 64-bit DecoderKey for EA11 or to calculate a 128-bit DecoderKey for EA07.

Table 40 – Data elements in DataBlock

No	Field	Description	Value	Bytes	Reference
1	SEP	Separator	0402 hex	2	
2	DKGA	DecoderKeyGeneratorAlgorithm	2 ASCII characters = “04” (3034 hex)	2	6.1.4
3	SEP	Separator	02 hex	1	
4	BDT	BaseDate	2 ASCII characters = “93” (3933 hex) or “14” (3134 hex) or “35” (3335 hex)	2	6.1.12
5	SEP	Separator	02 hex	1	
6	EA	EncryptionAlgorithm	2 ASCII characters	2	6.1.5
7	SEP	Separator	02 hex	1	
8	TI	TariffIndex	2 ASCII characters	2	6.1.7
9	SEP	Separator	000406 hex	3	
10	SGC	SupplyGroupCode	6 ASCII characters	6	6.1.6
11	SEP	Separator	01 hex	1	
12	KT	KeyType	1 ASCII character	1	6.1.9
13	SEP	Separator	01 hex	1	
14	KRN	KeyRevisionNumber	1 ASCII character	1	6.1.8
15	SEP	Separator	12 hex	1	
16	MeterPAN	MeterPAN	18 ASCII characters	18	6.1.2
17	L	Length of DK	4 byte (32 bit) integer	4	
			TOTAL	**49**	

For a DDTK or DUTK the actual designated DRN is used, but for a DCTK the DRN digits are set to zeros in the PANBlock, thus it always uses a fixed value of 0072700000000000.

Input parameters for a worked example are given in Table 41.

DRAFT

Table 41 – Input parameters for a worked example

Parameter	Value
VK	ABABABABABABABAB9494949494949494 01234567
MeterPAN	600727000000000009
KT	2
SGC	123456
TI	01
KRN	1
DKGA	04
BDT	93
EA	11

Construction of the DataBlock example is given in Table 42.

Table 42 – DataBlock example construction

Value	0402303402393302313102303100040631323334353601320131123630303732373 0 30303030303030303030303900000080

Construction of the DecoderKey example is given in Table 43.

Table 43 – DecoderKey construction example

128 bit key (EA = 11, L = 128)	28FEDCB88B215690E98EEAAB989E1C45 hex
64 bit key (EA = 07, L = 64)	A131DC9B419474BA hex

DRAFT

6.5.4 STA: EncryptionAlgorithm07

6.5.4.1 Encryption process

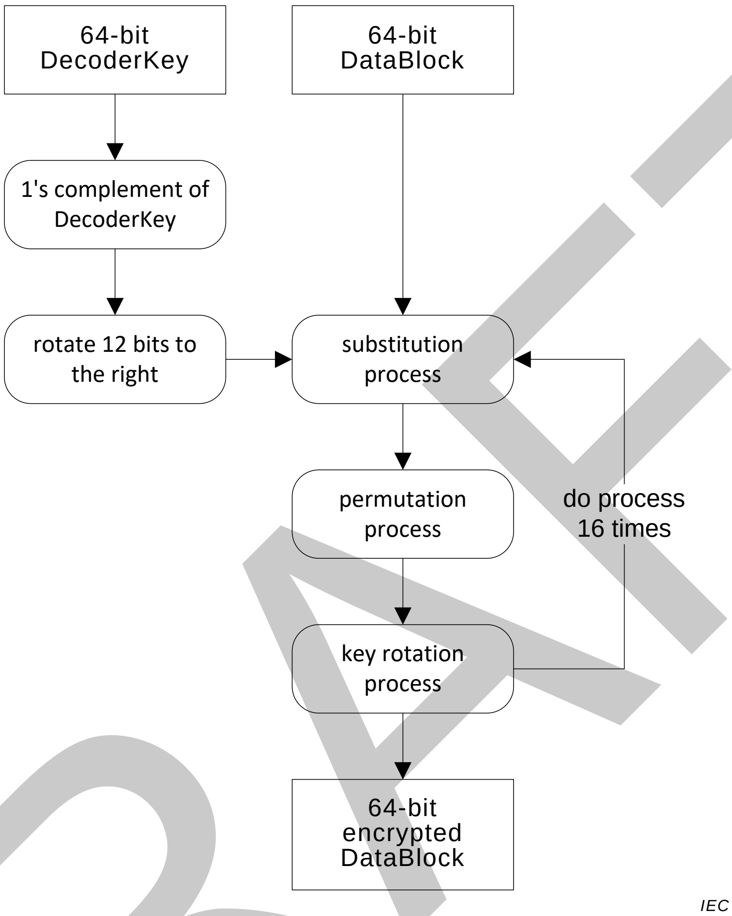


Figure 12 – STA: EncryptionAlgorithm07

The Standard Transfer Algorithm encryption process is shown in Figure 12, which comprises a key alignment process and 16 iterations of a substitution, permutation and key rotation process.

The POSApplicationProcess gives the appropriate directive by means of the EA code in the APDU.

6.5.4.2 Substitution process

The encryption substitution process is illustrated in Figure 13.

DRAFT

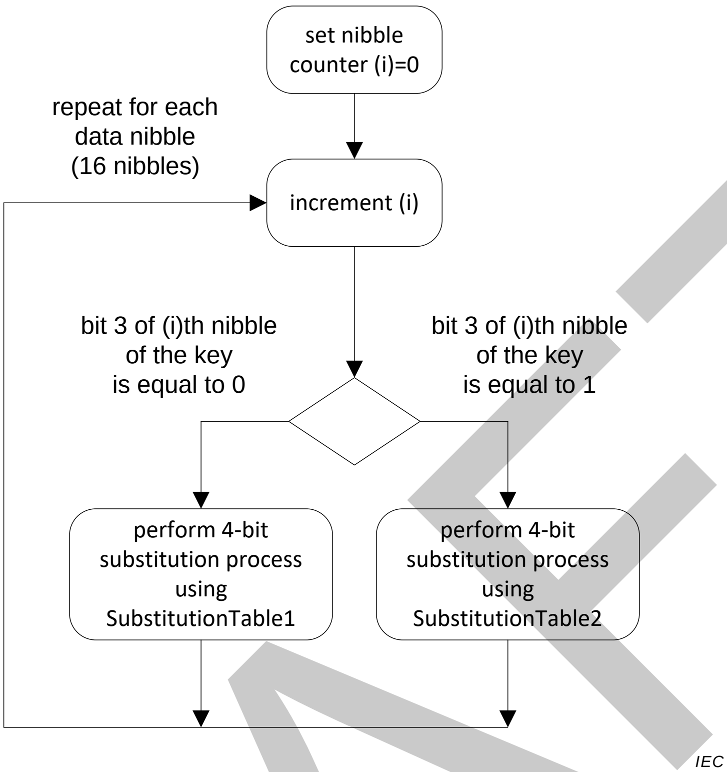


Figure 13 – STA encryption substitution process

There is a 4-bit substitution process for each of the 16 nibbles in the data stream. The substitution table used is one of two 16-value substitution tables and is dependent on the most significant bit setting of the corresponding nibble in the key. A sample substitution table is given in Table 44.

Table 44 – Sample substitution tables

SubstitutionTable1	12, 10, 8, 4, 3, 15, 0, 2, 14, 1, 5, 13, 6, 9, 7, 11
SubstitutionTable2	6, 9, 7, 4, 3, 10, 12, 14, 2, 13, 1, 15, 0, 11, 8, 5
NOTE This table contains only sample values (see Clause C.6 for access to table with actual values).	

The first entry in the substitution table corresponds to entry position 0 and the last to entry position 15.

Use the value of the data nibble as an index to an entry position in the substitution table; then replace the nibble value with the value from the substitution table found at that entry position. For example: if the value of the data nibble is 8 and we are using SubstitutionTable1, then the entry at position 8 is the value 14, thus replace the data nibble value with the value 14.

6.5.4.3 Permutation process

The encryption permutation process is illustrated in Figure 14.

DRAFT

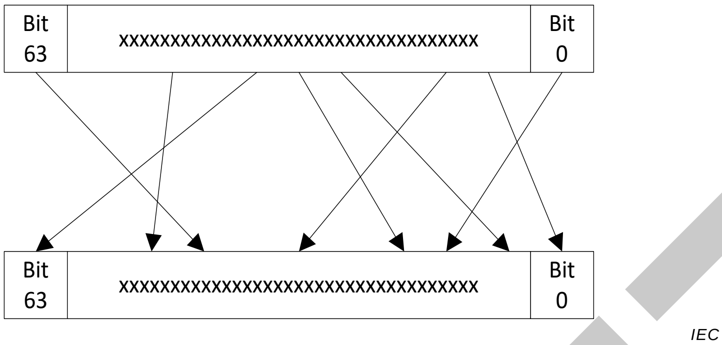


Figure 14 – STA encryption permutation process

A sample permutation table is given in Table 45.

Table 45 – Sample permutation table

PermutationTable3	29, 27, 34, 9, 16, 62, 55, 2, 40, 49, 38, 25, 33, 61, 30, 23, 1, 41, 21, 57, 42, 15, 5, 58, 19, 53, 22, 17, 48, 28, 24, 39, 3, 60, 36, 14, 11, 52, 54, 12, 31, 51, 10, 26, 0, 45, 37, 43, 44, 6, 59, 4, 7, 35, 56, 50, 13, 18, 32, 47, 46, 63, 20, 8
NOTE This table contains only sample values (see Clause C.6 for access to table with actual values).	

The first entry in the permutation table corresponds to the least significant bit position 0 in the DataBlock and the last entry to the most significant bit position 63 in the DataBlock.

Use the bit position of the source DataBlock as an index into the permutation table; then use the value found in the permutation table at that entry position as a pointer to the bit position in the destination DataBlock. For example: for the source DataBlock bit position 7 corresponds to the value 2 in the permutation table, thus the value of bit 7 from the source DataBlock is placed in bit position 2 in the destination DataBlock.

6.5.4.4 Key rotation process

The entire key is rotated one bit position to the left as illustrated in Figure 15.

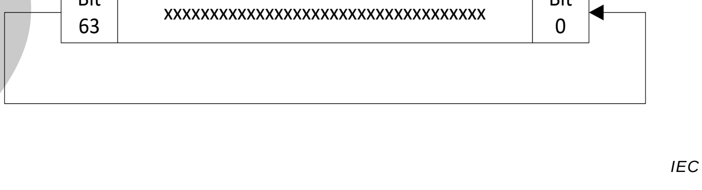


Figure 15 – STA encryption DecoderKey rotation process

6.5.4.5 Worked example to generate TokenData for a TransferCredit token using the STA

A worked example using the sample substitution and permutation tables is illustrated in Figure 16.

DRAFT

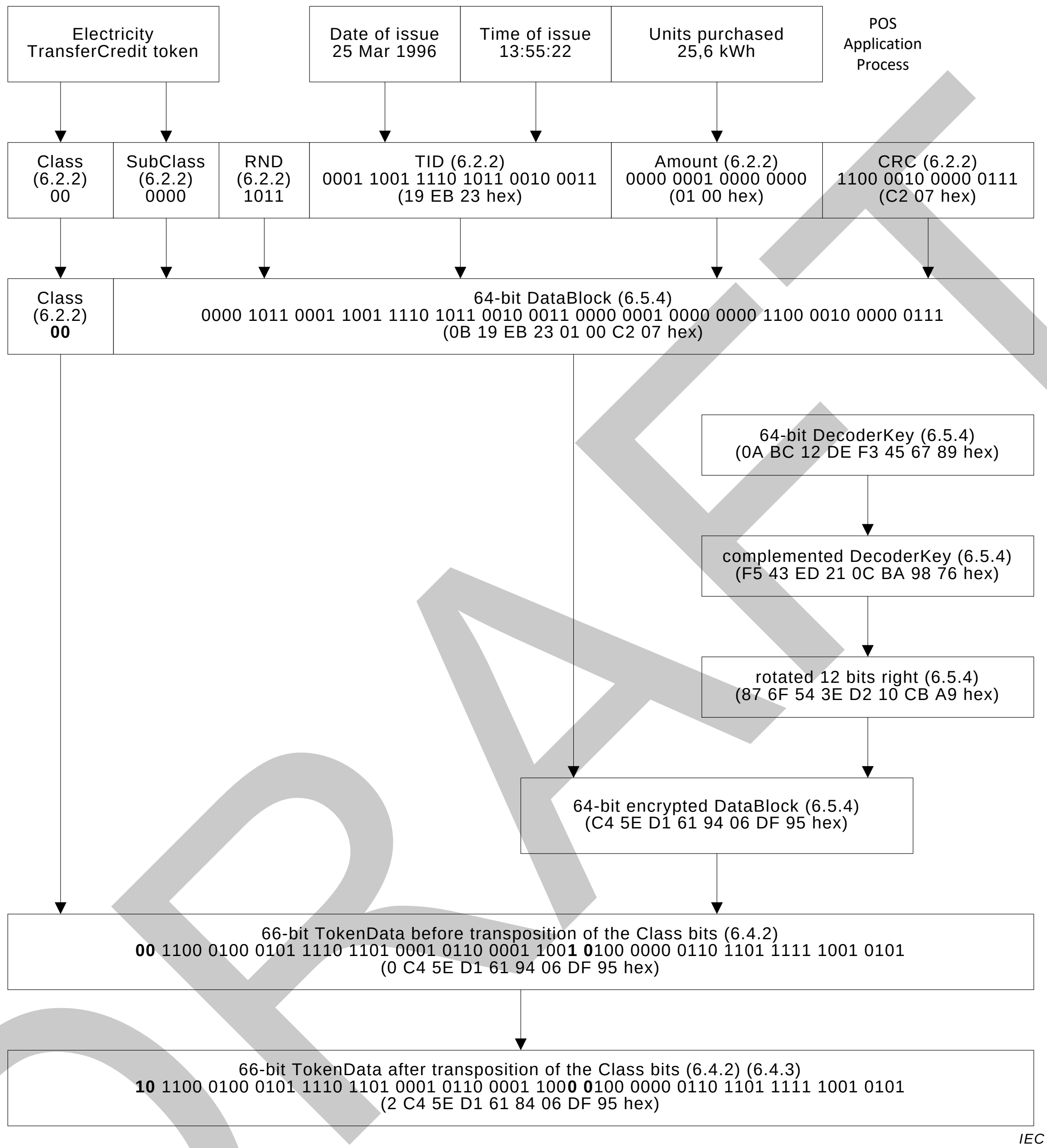


Figure 16 – STA encryption worked example for TransferCredit token

6.5.5 DEA: EncryptionAlgorithm09

This algorithm is deprecated and shall not be used in new products.

6.5.6 MISTY1: EncryptionAlgorithm11

6.5.6.1 Encryption process

The encryption process using the MISTY1 is shown in Figure 17.

DRAFT

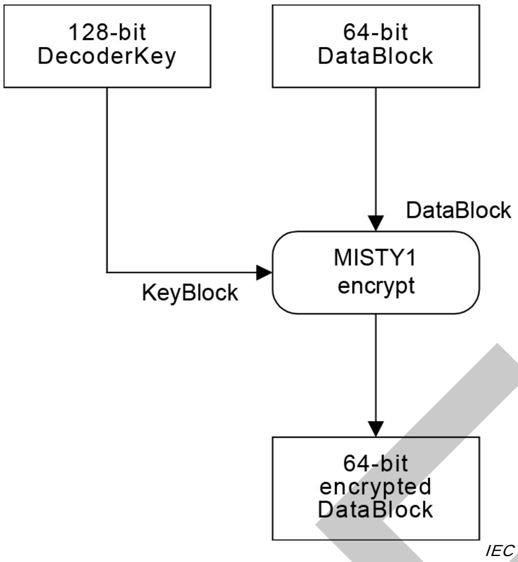


Figure 17 – MISTY1: EncryptionAlgorithm11

The MISTY1 is a 64-bit block cipher in accordance with ISO 18033-3. The POSApplicationProcess gives the appropriate directive by means of the EA code in the APDU.

The 128-bit DecoderKey is produced with DKGA04 as given in 6.5.3.6.

6.5.6.2 Worked example to generate TokenData for a TransferCredit token using MISTY1

A worked example using the MISTY1 encryption algorithm is illustrated in Figure 18.

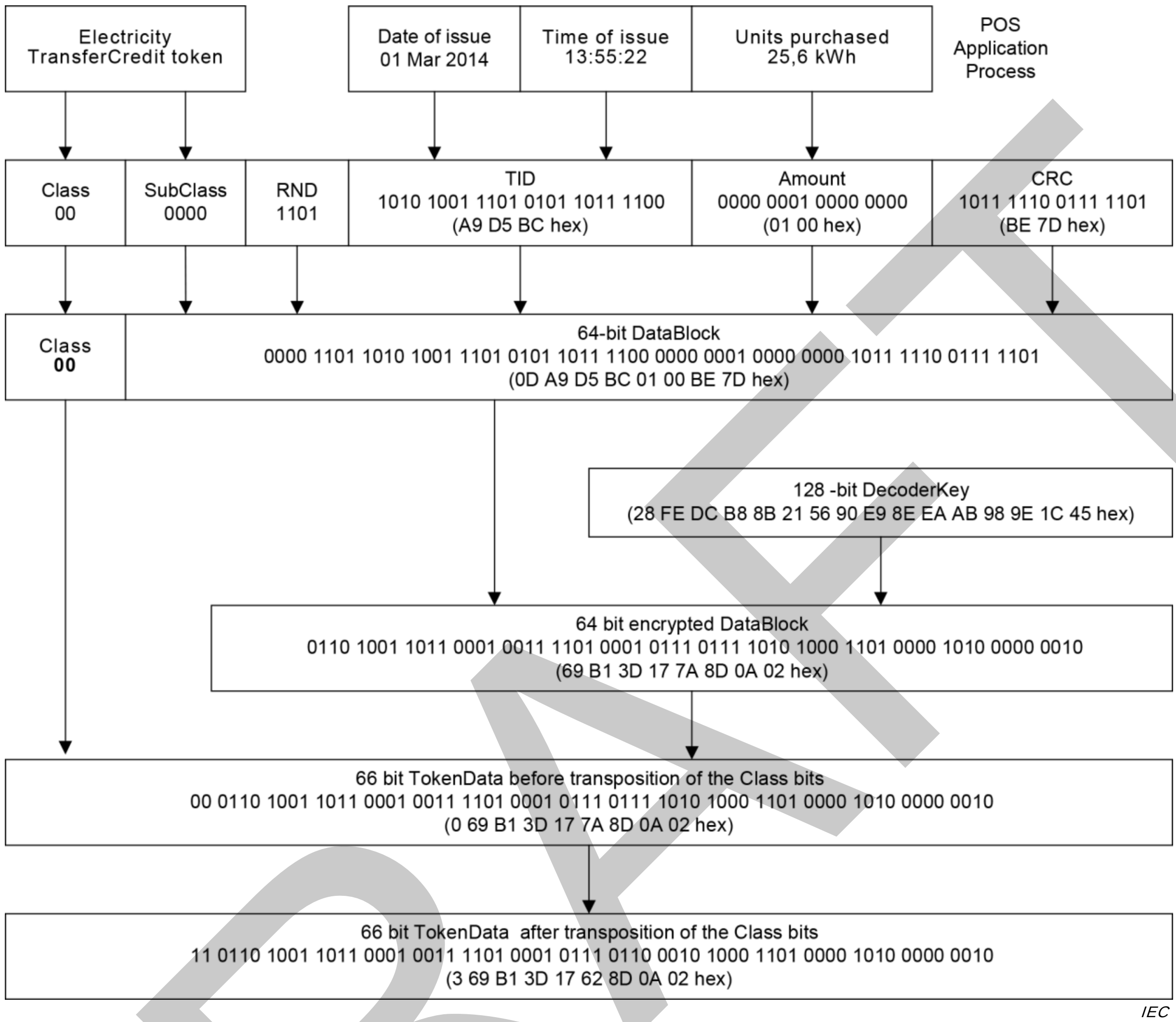


Figure 18 – MISTY1 encryption worked example for TransferCredit token

7 TokenCarriertoMeterInterface application layer protocol

7.1 APDU: ApplicationProtocolDataUnit

7.1.1 Data elements in the APDU

The APDU is the data interface between the MeterApplicationProcess and the application layer protocol and comprises the data elements given in Table 46.

DRAFT

Table 46 – Data elements in the APDU

Element	Context	Format	Reference
Token	The TokenData from the TCDU after decryption and processing; now presented to the MeterApplicationProcess in the APDU	66 bits	7.1.2
AuthenticationResult	Status indicator to the MeterApplicationProcess to convey the result from the initial authentication checks		7.1.3
ValidationResult	Status indicator to the MeterApplicationProcess to convey the result from the initial validation checks		7.1.4
TokenResult	Status indicator from the MeterApplicationProcess to convey the result after processing the token so that the application layer protocol can take the appropriate action		7.1.5

7.1.2 Token

The TokenData from the TCDU after decryption and processing; now presented to the MeterApplicationProcess in the APDU.

The actual 66-bit token as originally entered into the APDU by the MeterApplicationProcess. The MeterApplicationProcess is now able to process it further. See 6.2.1 for the detailed definition of this data element.

7.1.3 AuthenticationResult

A status indicator to tell the MeterApplicationProcess that the initial authentication checks (see 7.3.5) passed or failed, in order that the MeterApplicationProcess can respond appropriately. Possible values are given in Table 47.

Table 47 – Possible values for the AuthenticationResult

Value	Context	Format	Reference
Authentic	The authentication test passed or failed False if any one of the below error codes is indicated True if none of the below error codes is indicated	boolean	7.3.5
CRCError	The CRC value in the token is different to the CRC value as calculated from the data in the token	boolean	7.3.5
MfrCodeError	The MfrCode value in the Class 1 token does not match the MfrCode value for the Decoder	boolean	7.3.5

7.1.4 ValidationResult

A status indicator to tell the MeterApplicationProcess that the initial validation checks (see 7.3.7) passed or failed, in order that the MeterApplicationProcess can respond appropriately. Possible values are given in Table 48.

DRAFT

Table 48 – Possible values for the ValidationResult

Value	Context	Format	Reference
Valid	The Validation test passed or failed False if any one of the below error codes is indicated True if none of the below error codes is indicated	boolean	7.3.7
OldError	The TID value as recorded in the token is older than the oldest value of recorded values recorded in the memory store of the payment meter	boolean	7.3.7
UsedError	The TID value as recorded in the token is already recorded in the memory store of the payment meter	boolean	7.3.7
KeyExpiredError	The TID value as recorded in the token is larger than the KEN stored in the payment meter memory	boolean	7.3.7
DDTKError	The Decoder has a DDTK value in the DKR; a TransferCredit token may not be processed by the MeterApplicationProcess in accordance with the rules given in 6.5.2.3.3	boolean	7.3.7

7.1.5 TokenResult

After the MeterApplicationProcess has executed the instruction contained in the token, the TokenResult value reflects the outcome. The application layer protocol may then take the appropriate action to complete the token reading process, which may include accepting the token (and storing of the TID), rejection of the token, erasure of token data from the TokenCarrier, etc. Possible values are given in Table 49.

Table 49 – Possible values for the TokenResult

Value	Context	Format	Reference
Accept	The token was successfully processed False if any one of the below error codes is indicated True if none of the below error codes is indicated	boolean	8.2
1stKCT	The MeterApplicationProcess indicates that this is the Set1stSectionDecoderKey token of the set of key change tokens being read; the token is provisionally accepted	boolean	8.2
2ndKCT	The MeterApplicationProcess indicates that this is the Set2ndSectionDecoderKey token of the set of key change tokens being read; the token is provisionally accepted	boolean	8.2
3rdKCT	The MeterApplicationProcess indicates that this is the Set3rdSectionDecoderKey token of the set of key change tokens being read; the token is provisionally accepted	boolean	8.2
4thKCT	The MeterApplicationProcess indicates that this is the Set4thSectionDecoderKey token of the set of key change tokens being read; the token is provisionally accepted	boolean	8.2
OverflowError	The credit register in the payment meter would overflow if the token were to be accepted; the token is not accepted	boolean	8.2
KeyTypeError	The key may not be changed to this type in accordance with the key change rules given in 6.5.2.4.	boolean	8.2
FormatError	One or more data elements in the token does not comply with the required format for that element	boolean	8.2
RangeError	One or more data elements in the token have a value that is outside of the defined range of values defined in the application for that element	boolean	6.3
FunctionError	The particular function to execute the token is not available	boolean	8.2

DRAFT

7.2 APDUExtraction functions

7.2.1 Extraction process

The process of extracting the APDU from the TCDU is shown in Figure 19.

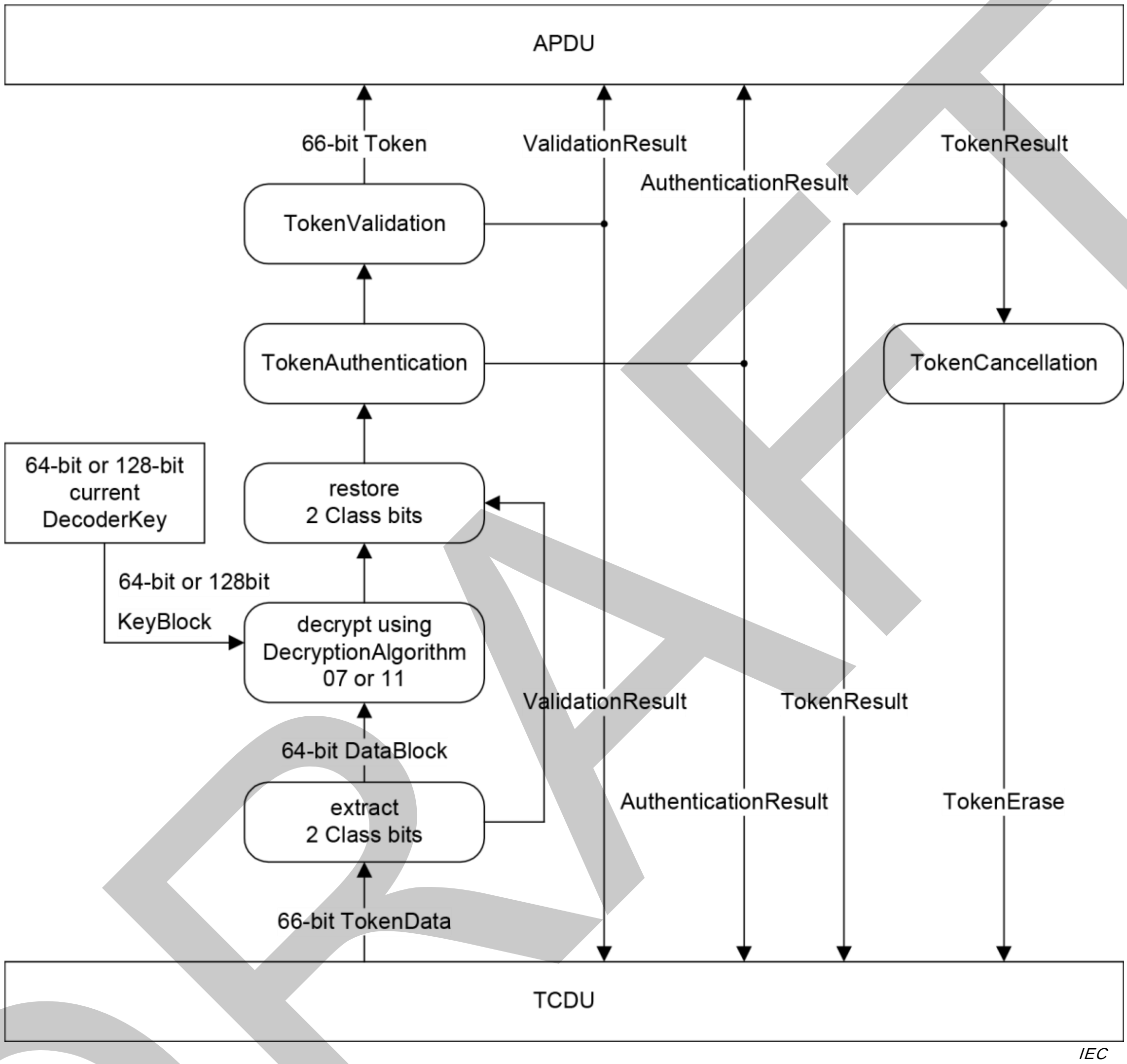


Figure 19 – APDUExtraction function

The APDUExtraction function extracts the 66-bit TokenData from the TCDU, decrypts and processes it before presenting the result in the APDU to the MeterApplicationProcess. It finally cancels and optionally causes the token data to be erased from the TokenCarrier in response to the result from the MeterApplicationProcess.

7.2.2 Extraction of the 2 Class bits

This function is used by other APDUExtraction functions (see 7.2.3 to 7.2.5). It removes the 2 Class bits from the 66-bit data stream to make a 64-bit number according to the method outlined in Figure 20 and is the inverse of 6.4.2.

The 66-bit number has its least significant bit in bit position 0 and its most significant bit in bit position 65. The 2-bit token Class value is extracted from bit positions 28 and 27. The values of bit positions 65 and 64 are relocated to bit positions 28 and 27. The most significant bit of the token Class comes from original bit position 28.

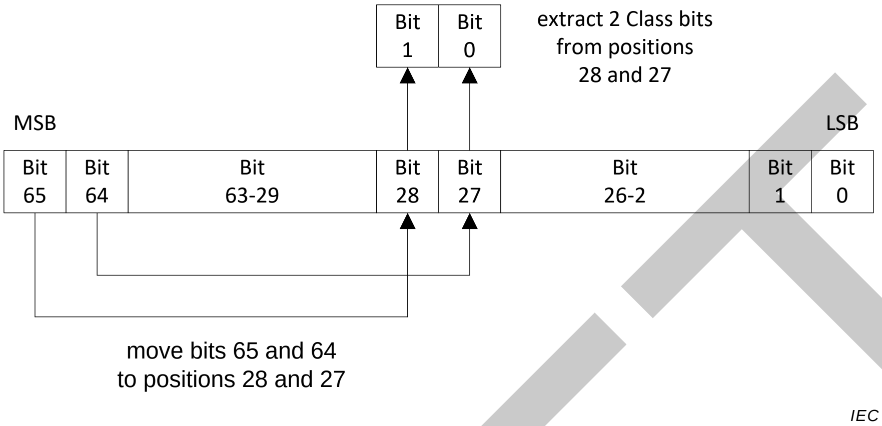


Figure 20 – Extraction of the 2 Class bits

Example: Extraction of the token Class = 01 (binary).

Extract the 2 Class bits from bit positions 28 and 27 (in bold):

00 0110 0101 0100 0011 0010 0001 0000 1001 100**0 1**111 0110 0101 0100 0011 0010 0001

Move bits 65 and 64 into bit positions 28 and 27 (in bold):

<u>00</u> 0110 0101 0100 0011 0010 0001 0000 1001 100**0 0**111 0110 0101 0100 0011 0010 0001

The resultant 64-bit binary number grouped in nibble (Bits 27 and 28 highlighted in bold):

0110 0101 0100 0011 0010 0001 0000 1001 100**0 0**111 0110 0101 0100 0011 0010 0001

7.2.3 APDUExtraction function for Class 0 and Class 2 tokens

This is the transfer function from the TCDU to the APDU and is applicable to all Class 0 and 2 tokens, except for the key change token set (see 7.2.5).

NOTE 1 The data elements in the APDU are defined in 7.1.1.

NOTE 2 The data elements in the TCDU are defined in each part of the IEC 62055-5x series physical layer protocol standard relevant to the specific TCT of interest.

The transfer function for Class 0 and Class 2 tokens is outlined as follows:

- the 2 Class bits are extracted from the 66-bit TokenData using the method in 7.2.2 to yield a 64-bit result, which is then presented to the decryption algorithm as its DataBlock input. Note that it is the responsibility of the POS to keep record of which specific decryption algorithm is in use in each particular payment meter (see 6.1.5 EA). The decryption algorithm and encryption algorithm are complementary and thus share the same EA code;
- the KeyBlock input for the decryption algorithm contains the current value of the DecoderKey, which is obtained from the DecoderKeyRegister in the payment meter secure memory;
- after decryption the 2 Class bits are again re-inserted into the 64-bit number to make a 66-bit number. The most significant bit of the 2 Class bits goes into bit position 65 and the least significant Class bit goes into bit position 64;
- the 66-bit token is authenticated in accordance with 7.3.5 and the result is indicated in the AuthenticationResult field of the APDU;
- the 66-bit token is validated in accordance with 7.3.7 and the result is indicated in the ValidationResult field of the APDU and the 66-bit token is placed in the Token field of the APDU;

- the MeterApplicationProcess processes the Token from the APDU and indicates the result in the TokenResult field of the APDU (see also 8.2). It is the responsibility of the MeterApplicationProcess to deal with display messages and indicators (see also 8.3) to the user and not the application layer protocol;
- if the TokenResult indicates Accept (see 7.1.5 and 8.2), then the Token is cancelled in accordance with 7.3.8 and the instruction is given in the TokenErase field of the TCDU to erase the data from the TokenCarrier.

NOTE 3 It is the responsibility of the physical layer protocol to decide whether the erase instruction is applicable or not in accordance with its specific implementation and TCT (see for example Clause 6 of IEC 62055-51:2007).

7.2.4 APDUExtraction function for Class 1 tokens

The APDUExtraction function for Class 1 tokens is identical to that of the Class 0 and Class 2 tokens, except that the decryption step is not performed.

7.2.5 APDUExtraction function for key change token set

This is the transfer function from the TCDU to the APDU and is applicable to the key change tokens.

NOTE 1 The data elements in the APDU are defined in 7.1.1.

NOTE 2 The data elements in the TCDU are defined in each part of the IEC 62055-5x series physical layer protocol standard relevant to the specific TCT of interest.

The transfer function for key change tokens is outlined as follows:

- the 2 Class bits are extracted from the 66-bit TokenData using the method in 7.2.2 to yield a 64-bit result, which is then presented to the decryption algorithm as its DataBlock input. Note that it is the responsibility of the POS to keep record of which specific decryption algorithm is in use in each particular payment meter (see 6.1.5 EA). The decryption algorithm and encryption algorithm are complementary and thus share the same EA code;
- the KeyBlock input for the decryption algorithm contains the current value of the DecoderKey, which is obtained from the DecoderKeyRegister in the payment meter secure memory;
- after decryption, the 2 Class bits are again re-inserted into the 64-bit number to make a 66-bit number. The most significant bit of the 2 Class bits goes into bit position 65 and the least significant Class bit goes into bit position 64;
- the 66-bit token is authenticated in accordance with 7.3.5 and the result is indicated in the AuthenticationResult field of the APDU;
- the 66-bit token is not validated in the application layer protocol, but only in the MeterApplicationProcess. The 66-bit token is placed in the Token field of the APDU;
- the MeterApplicationProcess processes the Token from the APDU and indicates the result in the TokenResult field of the APDU (see also 8.2). It is the responsibility of the MeterApplicationProcess to deal with display messages and indicators (see also 8.3) to the user and not the application layer protocol;
- if the TokenResult indicates 1stKCT,2ndKCT, 3rdKCT or 4thKCT (see 7.1.5 and 8.2) then the instruction to erase the data from the TokenCarrier is not given in the TokenErase field of the TCDU;
- if the TokenResult indicates Accept (see 7.1.5 and 8.2) then the instruction to erase the data from the TokenCarrier is given in the TokenErase field of the TCDU.

The key change tokens in the set may be entered in any order (see 8.9), but only the last one shall be erased.

NOTE 3 It is the responsibility of the physical layer protocol to decide whether the erase instruction is applicable or not, in accordance with its specific implementation and TCT (see for example Clause 6 of IEC 62055-51:2007).

DRAFT

7.3 Security functions

7.3.1 Key attributes and key changes

7.3.1.1 Key change requirements

The payment meter shall comply with the relevant requirements of 6.5.2, 7.3.1.2 and 7.3.1.3.

7.3.1.2 Key change processing without key expiry

The following defines the key change processing required if key expiry is not implemented in the payment meter:

– compare the KT value on the token against the KT value in the payment meter:
  - if KT values are equal, change the DecoderKeyRegister content, decoder KRN and payment meter TI to the corresponding new values on the token;
  - if KT values are not equal, validate KT rules (see 6.5.2.4):
    a) if key change is allowed, change the DecoderKeyRegister content, decoder KRN, decoder KT and payment meter TI to the corresponding new values on the token;
    b) if key change is not allowed, reject the key change operation.

7.3.1.3 Key change processing with key expiry

The following defines the key change processing required if key expiry is implemented in the payment meter:

– compare the token KT value against the decoder KT value:
  - if KT values are equal, change the DecoderKeyRegister content, decoder KEN, decoder KRN and payment meter TI to the corresponding token values;
  - if KT values are not equal, validate KT rules (see 6.5.2.4):
    a) if key change is allowed, change the DecoderKeyRegister content, decoder KEN, decoder KRN, decoder KT and payment meter TI to the corresponding token values;
    b) if key change is not allowed, reject the key change operation.

7.3.2 DKR: DecoderKeyRegister

The payment meter shall store the values given in Table 50 in secure non-volatile memory.

Table 50 – Values stored in the DKR

Value	Reference
DecoderKey	6.5.2.3.3, 6.5.3
TI	6.1.7
KRN	6.1.8
KT	6.1.9
KEN (optional)	6.1.10
SGC (optional)	6.1.6
The TI may be associated with a Tariff table that is managed outside of the domain of the payment meter. This implies that should a utility make use of the association, then the payment meter would require a key change each time that the customer is associated with a different tariff structure.	

In all cases where the payment meter provides configuration information, the KT shall be considered part of the KeyRevisionNumber information. The payment meter shall therefore always provide the KT information together with, or else directly after, the KRN information.

DRAFT

7.3.3 STA: DecryptionAlgorithm07

7.3.3.1 Decryption process

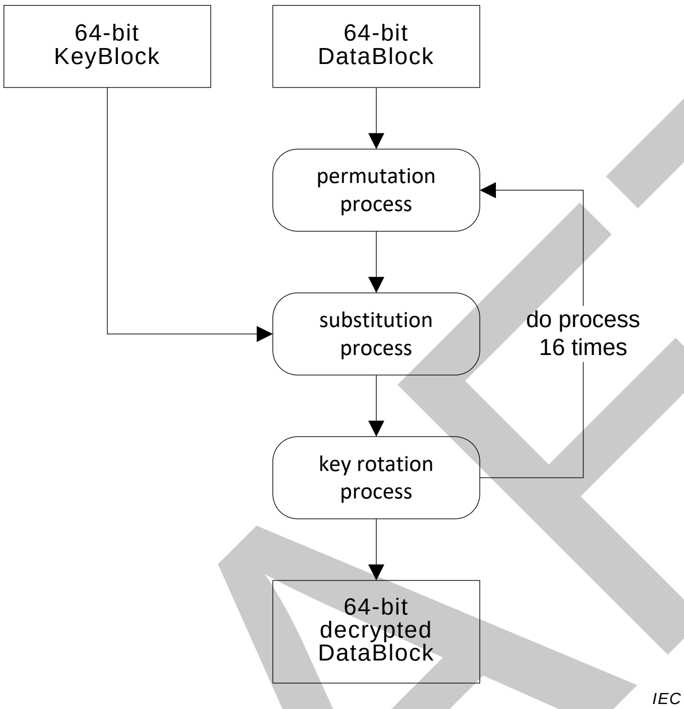


Figure 21 – STA DecryptionAlgorithm07

The Standard Transfer Algorithm decryption process is shown in Figure 21, which comprises a key alignment process and 16 iterations of a permutation, substitution and key rotation process.

The decryption algorithm and encryption algorithm are complementary and thus share the same EA code.

7.3.3.2 Permutation process

The decryption permutation process is illustrated in Figure 22.

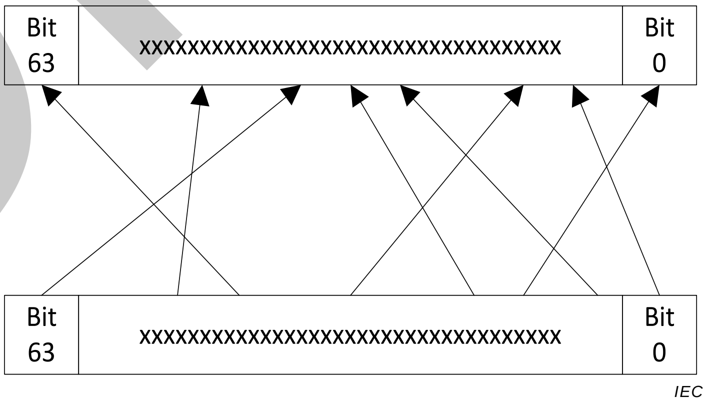


Figure 22 – STA decryption permutation process

DRAFT

A sample permutation table is given in Table 51.

Table 51 – Sample permutation table

PermutationTable4	44, 16, 7, 32, 51, 22, 49, 52, 63, 3, 42, 36, 39, 56, 35, 21, 4, 27, 57, 24, 62, 18, 26, 15, 30, 11, 43, 1, 29, 0, 14, 40, 58, 12, 2, 53, 34, 46, 10, 31, 8, 17, 20, 47, 48, 45, 60, 59, 28, 9, 55, 41, 37, 25, 38, 6, 54, 19, 23, 50, 33, 13, 5, 61
NOTE This table contains only sample values (see Clause C.6 for access to table with actual values).	

The first entry in the permutation table corresponds to the least significant bit position 0 in the DataBlock and the last entry to the most significant bit position 63 in the DataBlock.

Use the bit position of the source DataBlock as an index into the permutation table; then use the value found in the permutation table at that entry position as a pointer to the bit position in the destination DataBlock. For example: for the source DataBlock bit position 7 corresponds to the value 52 in the permutation table, thus the value of bit 7 from the source DataBlock is placed in bit position 52 in the destination DataBlock.

It can be seen that this gives the inverse result of the process in 6.5.4.3.

7.3.3.3 Substitution process

The decryption substitution process is illustrated in Figure 23.

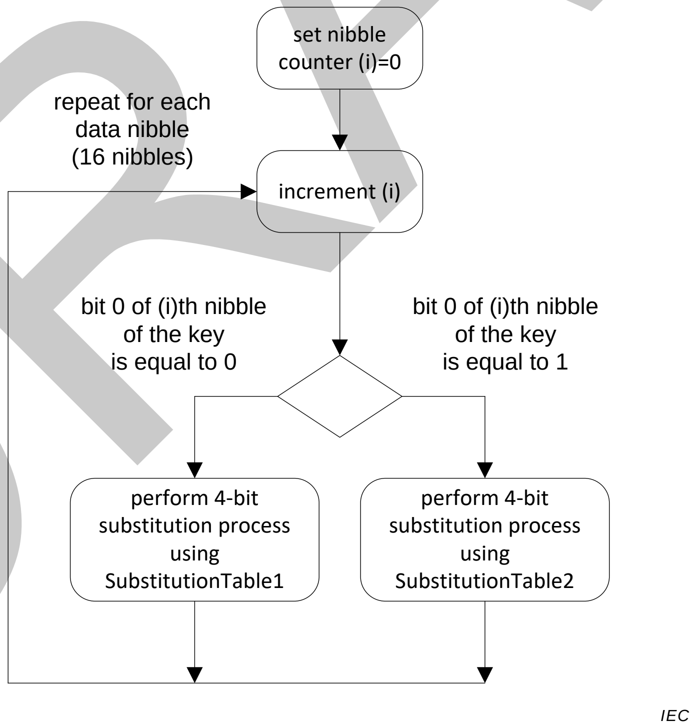


Figure 23 – STA decryption substitution process

There is a 4-bit substitution process for each of the 16 nibbles in the data stream. The substitution table used is one of two 16-value substitution tables and is dependent on the least significant bit setting of the corresponding nibble in the key. A sample substitution table is given in Table 52.

DRAFT

Table 52 – Sample substitution tables

SubstitutionTable1	12, 10, 8, 4, 3, 15, 0, 2, 14, 1, 5, 13, 6, 9, 7, 11
SubstitutionTable2	6, 9, 7, 4, 3, 10, 12, 14, 2, 13, 1, 15, 0, 11, 8, 5
NOTE This table contains only sample values (see Clause C.6 for access to table with actual values).	

The first entry in the substitution table corresponds to entry position 0 and the last to entry position 15.

Use the value of the data nibble as an index to an entry position in the substitution table; then replace the nibble value with the value from the substitution table found at that entry position. For example: if the value of the data nibble is 8 and we are using SubstitutionTable1, then the entry at position 8 is the value 14, thus replace the data nibble value with the value 14.

It can be seen that this gives the inverse result of the process in 6.5.4.2.

7.3.3.4 Key rotation process

The entire key is rotated one bit position to the right as illustrated in Figure 24.

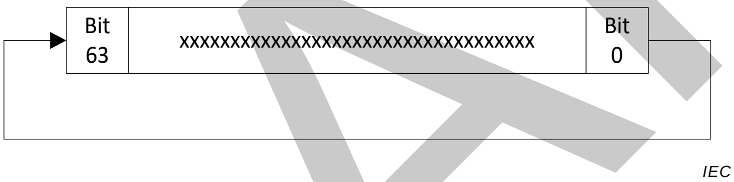


Figure 24 – STA decryption DecoderKey rotation process

7.3.3.5 Worked example to decrypt a TransferCredit token using the STA

A worked example using the sample substitution and permutation tables is illustrated in Figure 25.

DRAFT

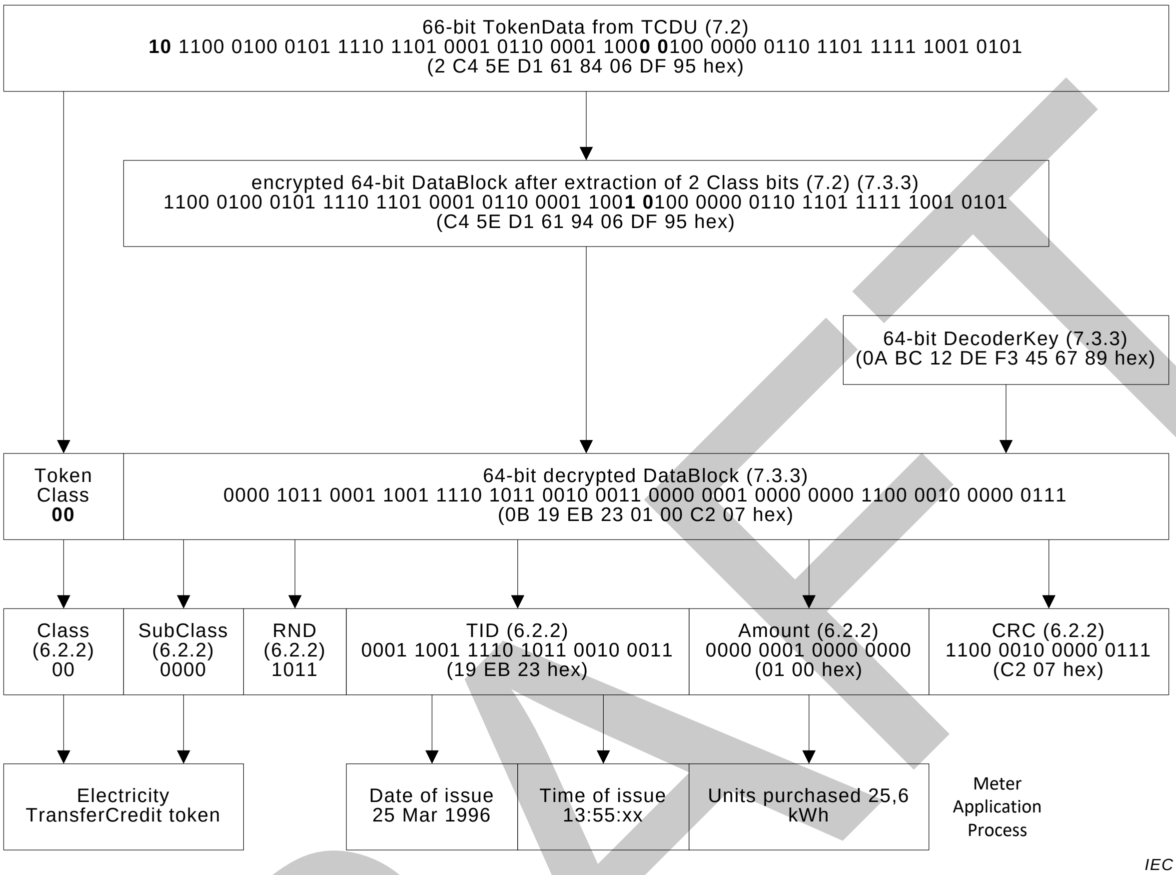


Figure 25 – STA decryption worked example for TransferCredit token

7.3.4 DEA: DecryptionAlgorithm09

This algorithm is deprecated and shall not be used in new products.

7.3.5 MISTY1: DecryptionAlgorithm11

7.3.5.1 Decryption process

The decryption process using the MISTY1 is shown in Figure 26.

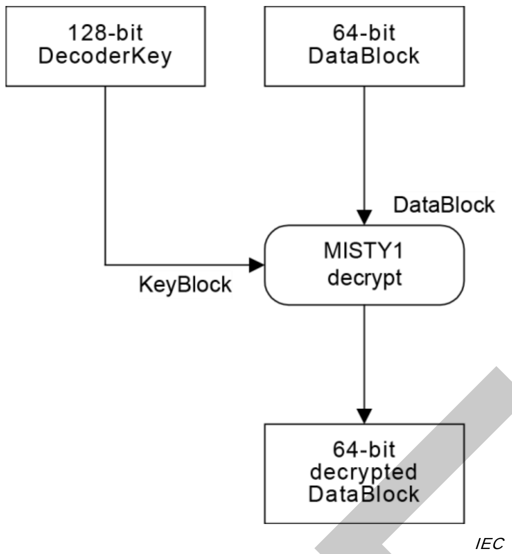


Figure 26 – STA DecryptionAlgorithm11

The decryption algorithm and encryption algorithm are complementary and thus share the same EA code.

7.3.5.2 Worked example to decrypt a TransferCredit token using the MISTY1

A worked example is illustrated in Figure 27.

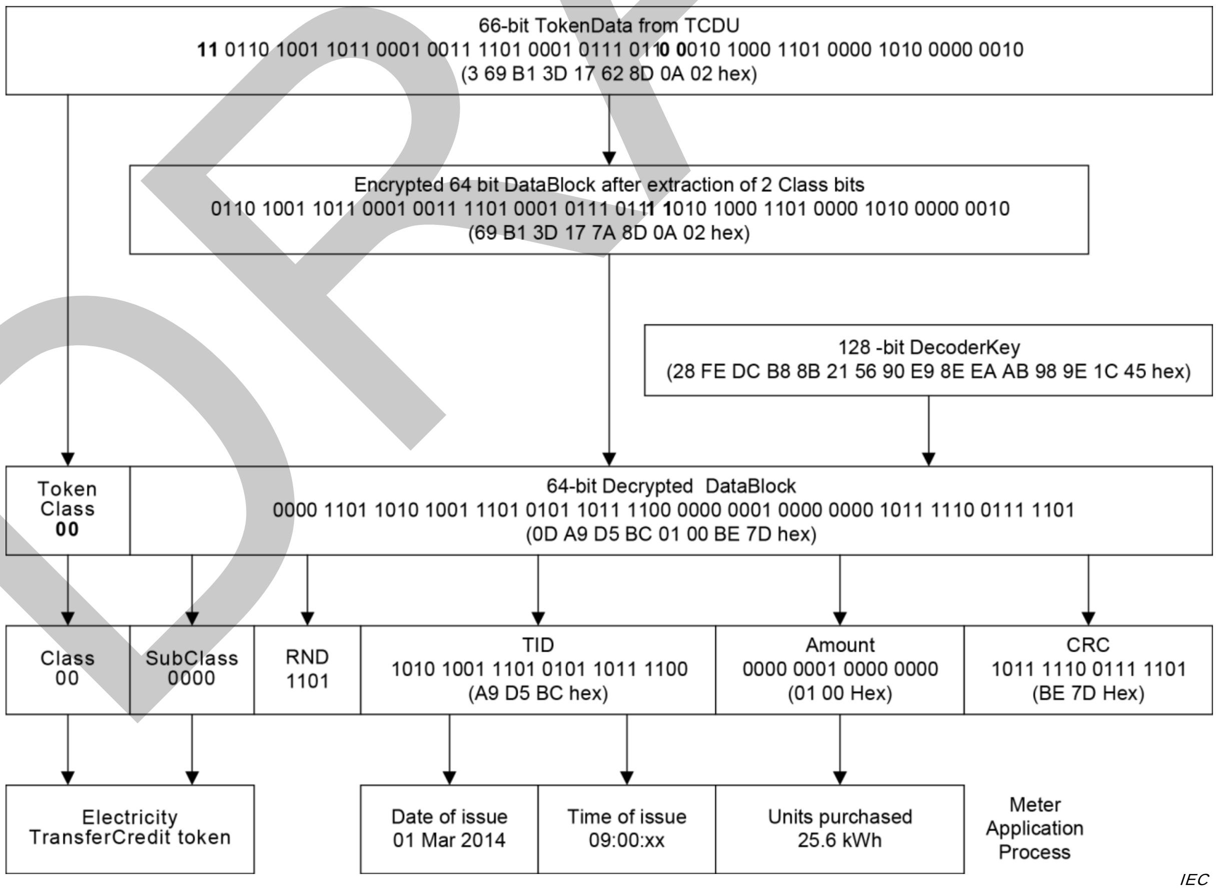


Figure 27 – MISTY1 decryption worked example for TransferCredit token

DRAFT

7.3.6 TokenAuthentication

Validating the CRC or the CRC_C checksum after decryption shall authenticate Class 0 and Class 2 tokens.

Validating the CRC and the MfrCode shall authenticate Class 1 tokens.

In the case of a Class 0 or a Class 2 token the AuthenticationResult status shall indicate Authentic when the following condition is met:

- the CRC or CRC_C checksum in the token has the same value as that calculated from the data elements in the token.

If the above condition is not met, then the AuthenticationResult status shall indicate CRCError.

In the case of a Class 1 token the AuthenticationResult status shall indicate Authentic when both of the following conditions are met:

- the CRC checksum in the token has the same value as that calculated from the data elements in the token;
- The MfrCode value in the token is the same as the MfrCode as defined in 6.2.3.

If any of the above conditions are not met, then the AuthenticationResult status shall indicate CRCError, or MfrCodeError, or both.

If the token cannot be authenticated, it shall be rejected in accordance with the requirements given in 8.2 and 8.3.

7.3.7 TokenValidation

Class 0 and Class 2 tokens shall primarily be validated against the TID encoded in the token, except for key change token set.

Key change tokens are validated by the MeterApplicationProcess once the payment meter has read all tokens and combined them into the new DecoderKey. See 8.2 for acceptance and rejection requirements of the key change tokens.

If key expiry is implemented in the payment meter, then the KEN stored in the payment meter shall also be used to validate tokens of Class 0 and Class 2 (see 6.5.2.6.), except for key change tokens.

A status of Valid shall be indicated if none of the following conditions are true:

- If a TID is received that has a value smaller than the smallest value of TID stored in the memory store (in other words, that was issued by a POS on a date before the earliest TID stored in the memory store), then such token containing this TID shall be rejected and indicate such condition as an OldError status (see 7.1.4);
- If a TID is received that is already stored in the memory store (see 7.3.8), the token shall be rejected and indicate such condition as a UsedError status (see 7.1.4);
- If key expiry is implemented in the payment meter and a TID is received that is greater than the KEN in the Decoder, the token shall be rejected and indicate such condition as a KeyExpiredError status (see 7.1.4);
- If a Class 0 token is presented to the Decoder with a DDTK value in the DKR, the token shall be rejected (see 6.5.2.3.3) and indicate such condition as a DDTKError status (see 7.1.4).

See also 8.2 and 8.3 for acceptance, rejection and indication requirements in the MeterApplicationProcess.

DRAFT

A payment meter loaded with a DDTK value shall accept all the relevant "non-meter-specific management tokens" (Class 1 tokens) as well as key change tokens encrypted under a DDTK.

7.3.8 TokenCancellation

Cancellation of a token shall be by means of storing the TID associated with that token in a secure non-volatile memory store in addition to erasure of the token data record from magnetic card token carriers (see 6.1.3 and 6.2.5 of IEC 62055-51:2007).

A time-based TID is used to uniquely identify each Class 0 and Class 2 token (except for the key change tokens). The payment meter shall store, in a secure non-volatile memory store, at least the last 50 TID values received.

If a valid token is received with a TID that has a value greater than the smallest value of TID value in the memory store and there is no available space in the memory store to store the received TID value, the payment meter shall accept this token, remove the smallest TID value (in other words, the oldest TID) from the memory store, and replace it with the new TID value.

If the payment meter accepts a key change token set, the TID memory store shall remain unchanged, unless the RolloverKeyChange (see 6.3.20) field specifies that the memory store shall be cleared.

The payment meter shall not accept tokens that were created prior to the date of manufacture or repair of the payment meter.

The manufacturer shall fill the TID memory store with values that indicate the date and time of manufacture or repair.

The payment meter shall read and process a token (as well as erase it when required) on a single insertion of the TokenCarrier without further action from the user.

All payment meters operating with a DCTK (see 6.5.2.3.1) shall erase token data (Class 0 and Class 2 tokens) from the TokenCarrier after successful transfer of the token data from the TokenCarrier to the payment meter, with the exception of the key change token data.

The following tokens shall not be erased:

- any token carrying a TID which is judged by the payment meter as being old;
- "non-meter-specific management tokens" of Class 1;
- the key change token set, except the last token entered.

The token in the key change token set, whichever is inserted last, shall be erased upon successful completion of the key change operation.

8 MeterApplicationProcess requirements

8.1 General requirements

In addition to the requirements given in Clause 8, the MeterApplicationProcess shall execute tokens in accordance with the definitions given in Clause 6 and Clause 7, and shall be further subject to the requirements given in IEC 62055-31 at all times, in particular the action of the load switch in response to remote replenishment of credit and the closing of the load switch from a remote location.

DRAFT

8.2 Token acceptance/rejection

An STS-compliant payment meter shall be capable of reading, interpreting and executing all of the categories of tokens successfully.

By default the payment meter shall still accept tokens when in the power limiting or tampered state, except when the purchase agreement between the manufacturer and the utility specifies otherwise.

Key change tokens are validated by the MeterApplicationProcess once the payment meter has read all tokens in the set and combined them into the new DecoderKey.

A token shall be accepted when all of the following conditions are true:

- AuthenticationResult indicates a status value of Authentic in the APDU (see 7.1.3);
- ValidationResult indicates a status value of Valid in the APDU (see 7.1.4);
- the token can be correctly interpreted and the instruction executed by the MeterApplicationProcess.

If all the above conditions are met, TokenResult (see 7.1.5) shall indicate Accept with the following exceptions:

- successful processing of the first entered token of a key change token set shall not indicate Accept, but it shall indicate 1stKCT, 2ndKCT, 3rdKCT or 4thKCT respectively for SubClass values 3, 4, 8 and 9, which indicators may be in any suitable format such as graphic icons or text and in any suitable language;
- successful processing of the last entered token of a key change token set shall indicate Accept.

The token shall be rejected and TokenResult shall not indicate Accept if any of the following conditions are true:

- AuthenticationResult does not indicate a status value of Authentic in the APDU (see 7.1.3);
- AuthenticationResult indicates a status value of CRCError in the APDU (see 7.1.3);
- AuthenticationResult indicates a status value of MfrCodeError in the APDU (see 7.1.3);
- ValidationResult does not indicate a status value of Valid in the APDU (see 7.1.4);
- ValidationResult indicates a status value of OldError in the APDU (see 7.1.4);
- ValidationResult indicates a status value of UsedError in the APDU (see 7.1.4);
- ValidationResult indicates a status value of KeyExpiredError in the APDU (see 7.1.4);
- ValidationResult indicates a status value of DDTKError in the APDU (see 7.1.4);
- In the case where completing the transaction execution of a TransferCredit token would cause the credit register in the payment meter to overflow, the TokenResult shall indicate OverflowError in the APDU (see 7.1.5) instead of Accept, the token shall be rejected and shall not be further processed;
- In the case where execution of a key change token would violate the key change rules as given in 6.5.2.4, the TokenResult shall indicate KeyTypeError in the APDU (see 7.1.5) instead of Accept, the token shall be rejected and shall not be further processed. See also 7.3.1 for further key change processing requirements;
- In the case where the structure of the token does not comply with the definitions given in 6.2, 6.3 or in the application for that token, the TokenResult shall indicate FormatError in the APDU (see 7.1.5) instead of Accept, the token shall be rejected and shall not be further processed;

DRAFT

- In the case where one or more data elements in the token have a value that is outside of the defined range of values defined in 6.2, 6.3 or in the the application for that element, the TokenResult shall indicate RangeError in the APDU (see 7.1.5) instead of Accept, the token shall be rejected and shall not be further processed;
- In the case where the particular function to execute the token is not implemented, the TokenResult shall indicate FunctionError in the APDU (see 7.1.5) instead of Accept, the token shall be rejected and shall not be further processed.

8.3 Display indicators and markings

The payment meter shall uniquely indicate the following conditions:

- the acceptance of a token (see 8.2);
- the rejection of a token (see 8.2);
- when a token is old (see 7.1.4);
- when a token has already been used, i.e. duplicate token (see 7.1.4);
- when the DecoderKey has expired (see 7.1.4);
- when a TransferCredit token is presented with a DDTK in the DKR (See 7.1.4);
- when the MeterApplicationProcess cannot execute the token (see 8.2);
- after a successful completion of a key change operation (see 8.2 and 8.9);
- whether accepting the credit on a token would cause the credit register to overflow (see 8.2).

Display indicators may be of any type and language (text, graphic, icon, etc.), but the type used for each display indication requirement shall be stated in the purchase agreement between the manufacturer and the utility.

The DRN and the EA code shall be marked on the part of the payment meter that contains the Decoder (see Clause 3) and shall be legible from the outside of the Decoder.

In the case where the Decoder part is separate from the user interface, then it shall be possible for the user to determine the DRN and the EA code from the user interface on demand by the push of a button, or entering a special code, or presentation of an InitiateMeterTest/Display token (see 6.2.3).

Indicators relating to the result of token entry shall only be displayed on the same user interface where the token was entered. In the case of a virtual token carrier for example, it is the task of the application layer protocol and the relevant physical layer protocol to feed back the ValidationResult, AuthenticationResult and TokenResult values via the same virtual token carrier interface.

8.4 TransferCredit tokens

See 6.2.2 for more detail on the structure of this token.

The credit value in the Amount field in the token shall be added to the available credit in the Accounting function in accordance with the specific implementation of the Accounting function and the service type as indicated by the SubClass field in the token.

8.5 InitiateMeterTest/Display tokens

See 6.2.3 for more detail on the structure of this token.

All payment meters shall support test number 0; if any of the incorporated tests are not supported the payment meter shall perform the subset of tests that are supported.

DRAFT

The relevant test shall be executed or the relevant information shall be displayed in accordance with the bit pattern in the Control field of the token.

When more than one output is required, for example for test number 0, the outputs shall be initiated in the order in which they are defined in 6.3.8. An optional test may be omitted if it is not implemented. A single test, for example test number 3, may provide more than one field of information.

Any optional tests not supported by the payment meter shall result in the rejection of the optional test token by the payment meter.

In the case where the SubClass value is in the range 6 to 15, the relevant test or display function shall be executed according to the manufacturer's specification, but the payment meter shall verify the MfrCode field value before such a token is accepted.

In the case where a payment meter has zero available credit which causes the load switch to be open, and the InitiateMeterTest/Display token may cause the load switch to operate into the closed state for the duration of the test. Some utilities may not want this condition to be allowed, while other utilities may want it. The action of the payment meter in response to this token shall be as agreed between the utility and the supplier and shall not form a normative part of this document.

8.6 SetMaximumPowerLimit tokens

See 6.2.4 for more detail on the structure of this token.

The present value of the maximum power limit register shall be replaced with the new limit.

The action of this function shall be agreed between the utility and the payment meter supplier.

NOTE 1 In a poly-phase payment meter this value is per phase.

NOTE 2 This function is not intended to be used as an over current protection mechanism, which requires adherence to other relevant standards.

8.7 ClearCredit tokens

See 6.2.5 for more detail on the structure of this token.

The available credit in the Accounting function shall be cleared to zero in accordance with the indicated value in the Register field of the token.

8.8 SetTariffRate tokens

See 6.2.6 for more detail on the structure of this token.

Reserved for future definition by the STS Association.

8.9 Key change tokens

See 6.2.7 and 6.2.8 for more detail on the structure of these tokens and token sets.

The present value of the DecoderKey shall be replaced with the new DecoderKey. The DecoderKey includes its associated attributes like KRN, KT, KEN, SGC and TI as defined in 7.3.2.

DRAFT

This action is subject to the successful receipt of all tokens in the token set. The payment meter shall have only one active DecoderKey at any stage of its operation. Dual DecoderKeys shall not be used.

It shall be possible to enter any token in the token set in any order to affect a successful key change.

It shall be possible to enter at least two other invalid tokens of any type and in any order, along with any one of the token set and still perform a successful key change.

It shall be possible to enter the same token from the token set more than once, if the key has not been changed already, and still perform a successful key change.

A time-out function shall be used to cancel a partially completed key change procedure after a duration of between 3 min and 10 min.

8.10 Set2ndSectionDecoderKey tokens

This subclause has been incorporated into 8.9.

8.11 ClearTamperCondition tokens

See 6.2.9 for more detail on the structure of this token.

The control status and indicator that indicates a tamper condition shall be reset to indicate a non-tamper condition. Any internal payment meter control process resultant from such a tamper condition shall also be cancelled.

8.12 SetMaximumPhasePowerUnbalanceLimit tokens

See 6.2.10 for more detail on the structure of this token.

The present value of the maximum phase unbalance power limit register shall be replaced with the new limit.

Implementation of this function in the payment meter is optional and the action of this function shall be agreed between the utility and the payment meter supplier.

NOTE This function is only applicable to poly-phase payment meters.

8.13 SetWaterMeterFactor

See 6.2.11 for more detail on the structure of this token.

The action of this token is reserved for future definition by the STS Association.

8.14 Class 2: Reserved for STS use tokens

See 6.2.12 for more detail on the structure of this token.

The payment meter shall reject these token types.

8.15 Class 2: Reserved for Proprietary use tokens

See 6.2.13 for more detail on the structure of this token.

DRAFT

The actions performed in the payment meter shall be in accordance with the manufacturer's specifications.

NOTE This document does not provide protection against collision between manufacturer uses of this token space.

8.16 Class 3: Reserved for STS use tokens

See 6.2.14 for more detail on the structure of this token.

The payment meter shall reject these token types.

9 KMS: KeyManagementSystem generic requirements

It is recognised that KMS requirements are essentially outside the scope of this document and the reader is therefore referred to relevant industry standards, some of which are listed in the Bibliography.

The STS Association has established well-proven codes of practice for the management of cryptographic keys within STS-compliant systems, utilising those industry standards, and it is therefore recommended that new systems implementing this document should follow the STS Association codes of practice.

By virtue of its Registration Authority status with IEC TC 13, the STS Association has undertaken to provide such certification services that are deemed necessary to ensure that key management systems comply with the relevant parts of this standard (see Clause C.1). For further guidelines on the functioning of a KeyManagementSystem as envisaged in this document, see Annex A.

10 Maintenance of STS entities and related services

10.1 General

See also Clause C.1 for more information relating to maintenance and support services.

The maintenance activity on certain STS entities requires a revision/amendment of this standard. Where this is the case, it is explicitly indicated as such.

Annex B and Annex C are not normative and any changes in these clauses due to maintenance activities would not require revision/amendment of this document, but may require appropriate amendments to other relevant specifications or COP.

The STS entities and services that require maintenance are given in Table 53.

Users of the STS refer to all parties that participate in the distribution and metering of utility services utilizing STS-compliant technology and also to the manufacturers and suppliers of such technology.

Access by STS users to STS entities and services as described in this document are thus regulated by the STS Association in accordance with appropriate rules and categorization of such users.

DRAFT

Table 53 – Entities/services requiring maintenance service

Entity/service	Definition origin	Responsible maintenance body	Reference
Product certification	Clause C.11	STSA/CA	10.2.1
DSN	6.1.2.3.3 C.4.4	manufacturer	10.2.2
RO	6.3.20	utility	10.2.3
TI	6.1.7	utility	10.2.4
TID	6.3.5.1	utility	10.2.5
SpecialReservedTokenIdentifier	6.3.5.2 Clause C.5	utility	10.2.6
MfrCode	6.1.2.3.2 C.4.3	STSA	10.2.7
Substitution tables	6.5.4.2 7.3.3.3 Clause C.6	STSA	10.2.8
Permutation tables	6.5.4.3 7.3.3.2 Clause C.6	STSA	10.2.9
SGC	6.1.6 C.2.2	STSA/KMC	10.2.10
VendingKey	6.5.2.2 Clause 9 C.3.2	STSA/KMC	10.2.11
KRN	6.1.8 6.5.2.5	STSA/KMC	10.2.12
KT	6.1.9 6.5.2 Table 37	STSA/KMC	10.2.13
KEN	6.1.10 6.5.2.6 C.3.4	STSA/KMC	10.2.14
CERT	Annex B Table B.1	STSA/KMC	10.2.15
CC	Annex B Table B.2	STSA/KMC	10.2.16
UC	Annex B Table B.2	STSA/KMC	10.2.17
KMCID	Annex B Table B.2	STSA/KMC	10.2.18
CMID	Annex B Table B.2	manufacturer/KMC	10.2.19
IIN	6.1.2.2 C.4.2	STSA	10.3.1

DRAFT

Entity/service	Definition origin	Responsible maintenance body	Reference
TCT	6.1.3 Table 5	STSA/IEC	10.3.2
DKGA	6.1.4 Table 6	STSA/IEC	10.3.3
EA	6.1.5 Table 7	STSA/IEC	10.3.4
TokenClass	6.3.2 Table 14 Table 15	STSA/IEC	10.3.5
TokenSubClass	6.3.3 Table 15	STSA/IEC	10.3.6
InitiateMeterTest/Display ControlField	6.3.8 Table 27	STSA/IEC	10.3.7
RegisterToClear	6.3.13 Table 28	STSA/IEC	10.3.8
STS base date	6.3.5.1	STSA/IEC	10.3.9
Rate	6.3.11	STSA/IEC	10.3.10
WMFactor	6.3.12	STSA/IEC	10.3.11
MFO	5.5	STSA/(IEC)	10.3.12
FOIN	5.5 Clause C.9	STSA/(IEC)	10.3.13
Companion Specification	5.5 Clause C.9	STSA/(IEC)	10.3.14

10.2 Operations

10.2.1 Product certification maintenance

The STS Association, as a registered Registration Authority with the IEC, shall ensure access to product certification services, subject to legal requirements ruling at the time.

It shall also ensure that such service providers are duly accredited and authorized to provide this service and that they comply with the requirements of this document and any other relevant COP or specification.

10.2.2 DSN maintenance

The payment meter manufacturer is in complete control of his allocated range of DSN values (within his allocated MfrCode domain) and it thus requires no further maintenance.

10.2.3 RO maintenance

The utility shall manage the operational use of this data element in conjunction with the STS BaseDate.

10.2.4 TI maintenance

The utility shall manage the operational use of this element.

DRAFT

10.2.5 TID maintenance

The utility shall manage the operational use of this data element by means of appropriate programming of the token vending or POS systems.

10.2.6 SpecialReservedTokenIdentifier maintenance

The utility shall manage the operational use of this data element by means of appropriate programming of the token vending or POS systems.

10.2.7 MfrCode maintenance

The STS Association, as a registered Registration Authority with the IEC, shall provide the service of allocating MfrCode values to payment meter manufacturers and making the list of allocated MfrCode values available to users of the STS upon request.

10.2.8 Substitution tables maintenance

The STS Association, as a registered Registration Authority with the IEC, shall provide the service of making the actual values for Table 44 and Table 52 available to users of the STS upon request.

10.2.9 Permutation tables maintenance

The STS Association, as a registered Registration Authority with the IEC, shall provide the service of making the actual values for Table 45 and Table 51 available to users of the STS upon request.

10.2.10 SGC maintenance

The STS Association, as a registered Registration Authority with the IEC, shall ensure access to SGC allocation services to users of the STS and that SGC values are globally unique. Such services are typically provided by a KMC.

10.2.11 VendingKey maintenance

The STS Association, as a registered Registration Authority with the IEC, shall ensure access to VendingKey allocation services to users of the STS, that VendingKey values are globally unique and that VendingKey values are made available between KMC service providers. Such services are typically provided by a KMC.

The STS Association shall also ensure the compliance of such service providers to the requirements and recommendations given in this document and any other relevant COP or specification.

10.2.12 KRN maintenance

This element is intrinsically coupled to the VendingKey and is managed by the KMC service provider, subject to the same conditions as for VendingKey maintenance.

10.2.13 KT maintenance

This element is intrinsically coupled to the VendingKey and is managed by the KMC service provider, subject to the same conditions as for VendingKey maintenance.

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 shall administer any further additions to the range of KeyType values as given in Table 37.

DRAFT

The process shall follow the standard procedures for submission of new work item proposals, as instituted by these organisations.

An additional KeyType definition shall require a revision/amendment of this document.

10.2.14 KEN maintenance

This element is intrinsically coupled to the VendingKey and is managed by the KMC service provider, subject to the same conditions as for VendingKey maintenance.

10.2.15 CERT maintenance

The KMC service provider is exclusively in control of this data element as it forms an intrinsic part of its key management operations.

The STS Association, as a registered Registration Authority with the IEC, shall ensure that KMC service providers comply with the requirements of this document and any other relevant COP.

10.2.16 CC maintenance

The STS Association, as a registered Registration Authority with the IEC, shall ensure access to CC allocation services to users of the STS and that CC values are globally unique. Such services are typically provided by a KMC.

10.2.17 UC maintenance

The STS Association, as a registered Registration Authority with the IEC, shall ensure access to UC allocation services to users of the STS and that UC values are globally unique. A KMC typically provides such services.

10.2.18 KMCID maintenance

The STS Association, as a registered Registration Authority with the IEC, shall ensure access to KMCID allocation services to users of the STS and that KMCID values are globally unique. The STS Association typically provides such services.

10.2.19 CMID maintenance

The CM manufacturer is in complete control of allocating CMID values to his manufactured CM devices and there is no service in place to ensure uniqueness of this data element.

Once a particular CM is registered in an STS system (typically with a KMC service provider), then the CMID is simply recorded for reference purposes and no further maintenance service on this data element is required.

10.3 Standardisation

10.3.1 IIN maintenance

This document defines two constant values for electricity payment meters worldwide.

Different values of IIN are reserved for future definition by the STS Association.

Any changes to the rules as defined in this document would require a revision/amendment of this document.

DRAFT

10.3.2 TCT maintenance

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 shall administer any further additions to the range of TCT values given in Table 5.

The process shall follow the standard procedures for submission of new work item proposals, as instituted by these organisations.

An additional entry to Table 5 shall require a revision/amendment of this document and a new part in the IEC 62055-5x series.

10.3.3 DKGA maintenance

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 shall administer any further additions to the range of DKGA values given in Table 6.

The process shall follow the standard procedures for submission of new work item proposals, as instituted by these organisations.

An additional entry to Table 6 shall require a revision/amendment of this document.

10.3.4 EA maintenance

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 shall administer any further additions to the range of EA values given in Table 7.

The process shall follow the standard procedures for submission of new work item proposals, as instituted by these organisations.

An additional entry to Table 7 shall require a revision/amendment of this document.

10.3.5 TokenClass maintenance

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 shall administer any further additions to the range of TokenClass values as given in Table 14 and Table 15.

The process shall follow the standard procedures for submission of new work item proposals, as instituted by these organisations.

An additional TokenClass definition shall require a revision/amendment of this document.

10.3.6 TokenSubClass maintenance

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 shall administer any further additions to the range of TokenSubClass values as given in Table 15.

The process shall follow the standard procedures for submission of new work item proposals, as instituted by these organisations.

An additional TokenSubClass definition shall require a revision/amendment of this document.

10.3.7 InitiateMeterTest/DisplayControlField maintenance

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 shall administer any further additions to the range of InitiateMeterTest/DisplayControlField values given in Table 27.

DRAFT

The process shall follow the standard procedures for submission of new work item proposals, as instituted by these organisations.

An additional InitiateMeterTest/DisplayControlField value shall require a revision/amendment of this document.

10.3.8 RegisterToClear maintenance

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 shall administer any further additions to the range of RegisterToClear values given in Table 28.

The process shall follow the standard procedures for submission of new work item proposals, as instituted by these organisations.

An additional RegisterToClear value shall require a revision/amendment of this document.

10.3.9 STS BaseDate maintenance

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 shall administer any changes to the STS base date.

The process shall follow the standard procedures for submission of new work item proposals, as instituted by these organisations.

A change in the STS BaseDate value shall require a revision/amendment of this document.

10.3.10 Rate maintenance

This data element is presently reserved for future definition by the STS Association.

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 shall administer any changes to the definition of the Rate data element.

The process shall follow the standard procedures for submission of new work item proposals, as instituted by these organisations.

A change in definition of the Rate data element shall require a revision/amendment of this document.

10.3.11 WMFactor maintenance

This data element is presently reserved for future definition by the STS Association.

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 shall administer any changes to the definition of the WMFactor data element.

The process shall follow the standard procedures for submission of new work item proposals, as instituted by these organisations.

A change in definition of the WMFactor data element shall require a revision/amendment of this document.

10.3.12 MFO maintenance

Definitions of MFO instances are presently outside the normative domain of this document and are mentioned purely on an informative basis.

DRAFT

The STS Association exclusively administers the definition of MFO instances following its own internal standard procedures for submission of new work item proposals.

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 may in the future propose these MFO instances to the IEC for development into international standards, which shall follow the standard procedures for submission of new work item proposals, as instituted by the IEC.

10.3.13 FOIN maintenance

Allocation and assignment of FOIN values are presently outside the normative domain of this document and are mentioned purely on an informative basis.

The STS Association exclusively administers the allocation and assignment of FOIN values in conjunction with the registration of MFO instances as companion specifications.

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 may in the future propose these FOIN values to the IEC for development into international standards, which shall follow the standard procedures for submission of new work item proposals, as instituted by the IEC.

10.3.14 Companion specification maintenance

Development of companion specifications is presently outside the normative domain of this document and is mentioned purely on an informative basis.

The STS Association exclusively administers the development of companion specifications in conjunction with registration of MFO instances and assignment of FOIN values.

The STS Association in liaison partnership with Working Group 15 of IEC TC 13 may in the future propose these companion specifications to the IEC for development into international standards, which shall follow the standard procedures for submission of new work item proposals, as instituted by the IEC.

DRAFT

Annex A
(informative)

Guidelines for a KeyManagementSystem (KMS)

This informative Annex provides general guidelines for the implementation of a KMS for the management of the cryptographic keys as required to satisfy the normative requirements of this document and uses techniques, processes and procedures as prescribed by the NIST and FIPS standards. It should be noted that the deployment of such a KMS could possibly be in conflict with some country-specific or regional-specific regulatory requirements for the management of cryptographic keys for application in utility distribution or metering systems. It is outside of the scope of this Annex to deal with such possible conflicts.

An entity relation and interaction diagram is shown in Figure A.1.

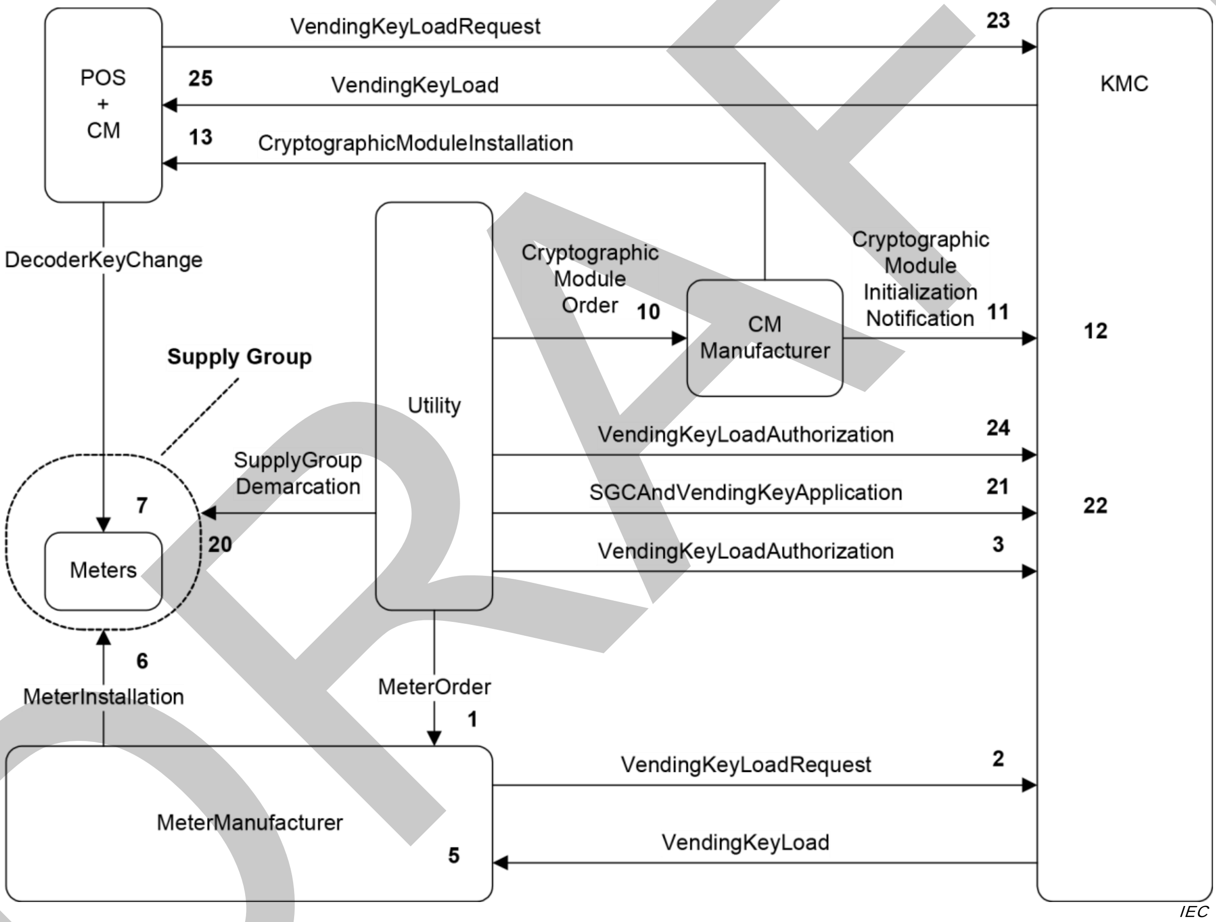


Figure A.1 – KeyManagementSystem and interactive relationships between entities

The entities that play a role in the KMS processes are given in Table A.1.

DRAFT

Table A.1 – Entities that participate in KMS processes

Entity	Role / Name
Utility	Supplier of a service such as electricity
MeterManufacturer	Manufacturer of payment meters/ decoder devices
CMManufacturer	Manufacturer of cryptographic modules
KMC	KeyManagementCentre
CM	CryptographicModule
POS	PointOfSale
Meter	Payment meter

The payment meter processes and DecoderKey processes are given in Table A.2.

Table A.2 – Processes surrounding the payment meter and DecoderKey

Process Number	Context
1	MeterOrder Utility places an order for payment meters with the MeterManufacturer. The order will stipulate that the payment meters are loaded with DDTK, DUTK or DCTK values for the specified SGC
2	VendingKeyLoadRequest MeterManufacturer requests the VendingKey (VUDK or VCDK) for the specific SGC, if required, from the KMC, else he uses his own allocated VDDK (see 6.5.2.2) or the VDDK owned by the Utility
3	VendingKeyLoadAuthorization The Utility authorizes the KMC to load the requested VendingKey values down to the MeterManufacturer
4	VendingKeyLoad The requested VendingKey values are loaded into the MeterManufacturer’s STS-certified secure manufacturing equipment
5	DecoderKeyLoad The MeterManufacturer generates the DDTK, DUTK or DCTK values from the VDDK, VUDK or VCDK values in accordance with the payment meter order and loads these into the payment meter (see 6.5.3)
6	MeterInstallation The payment meters are delivered to the Utility and installed in the demarcated SupplyGroup
7	DecoderKeyChange If so required the DecoderKey value may be changed by vending KeyChangeTokens from the POS equipment (see 6.2.7 and 6.2.8). See also processes 23 to 25 below regarding VendingKey loading

The CryptographicModule processes are given in Table A.3.

DRAFT

Table A.3 – Processes surrounding the CryptographicModule

Process Number	Context
10	CryptographicModuleOrder The Utility (or POS manufacturer) places an order for a cryptographic module with a cryptographic module manufacturer
11	CryptographicModuleInitialisationNotification The CMManufacturer initialises the CryptographicModule with public and private key values, which will subsequently be utilized for securely distributing VendingKey values from the KMC to the CryptographicModule. The certified public key values and associated parameters are sent to the KMC for registration of the new CryptographicModule.
12	CryptographicModuleAuthenticationAndRegistration The KMC registers CryptographicModule parameters and certified public key values in the KMC, which will subsequently be utilized for securely distributing VendingKey values from the KMC to the CryptographicModule
13	CryptographicModuleInstallation The CryptographicModule is installed and is ready for loading of VendingKey values from the KMC typically using KeyLoadFiles (see KLF in Annex B)

The SGC and VendingKey processes are given in Table A.4.

Table A.4 – Processes surrounding the SGC and VendingKey

Process Number	Context
20	SupplyGroupDemarcation The Utility supplies electricity to a defined group of its customers. It decides the size and boundaries of the group based on security risk and revenue protection considerations, geographical location and network logistical characteristics
21	SGCAndVendingKeyApplication The Utility makes application to the KMC for a SGC of specified type (unique, common or default) and associated VendingKey of a specified type (VUDK, VCDK or VDDK; see 6.5.3)
22	SGCAndVendingKeyAllocation The KMC allocates a SGC and an associated secret VendingKey of the required KT to the applicant and stores the elements in its records
23	VendingKeyLoadRequest POS operator requests the VendingKey value (VDDK, VUDK or VCDK) for the specific SGC from the KMC that will allow him to vend to payment meters loaded with the associated DecoderKey value (DDTK, DUTK or DCTK)
24	VendingKeyLoadAuthorization The Utility authorizes the KMC to load the requested VendingKey values (VUDK, VCDK or VDDK). Alternatively the MeterManufacturer authorizes the KMC to load the requested VDDK value
25	VendingKeyLoad The requested VendingKey values are loaded into the CryptographicModule that will be used by the POS equipment to generate tokens for the payment meters in the SupplyGroup

The mandatory requirements for a KeyManagementSystem are specified in Clause 9.

See also Clause C.3 Code of practice for more information regarding the management of VendingKeys.

DRAFT

See also C.3.2.1 Code of practice for more information regarding the SGC demarcation guidelines.

See also Annex B for more information regarding entities and identifiers in an STS-compliant system.

See also Clause 10 for the maintenance of the STS entities and related services.

DRAFT

Annex B
(informative)

Entities and identifiers in an STS-compliant system

Entities and relevant identifiers deployed in an STS-compliant system are shown in Figure B.1.

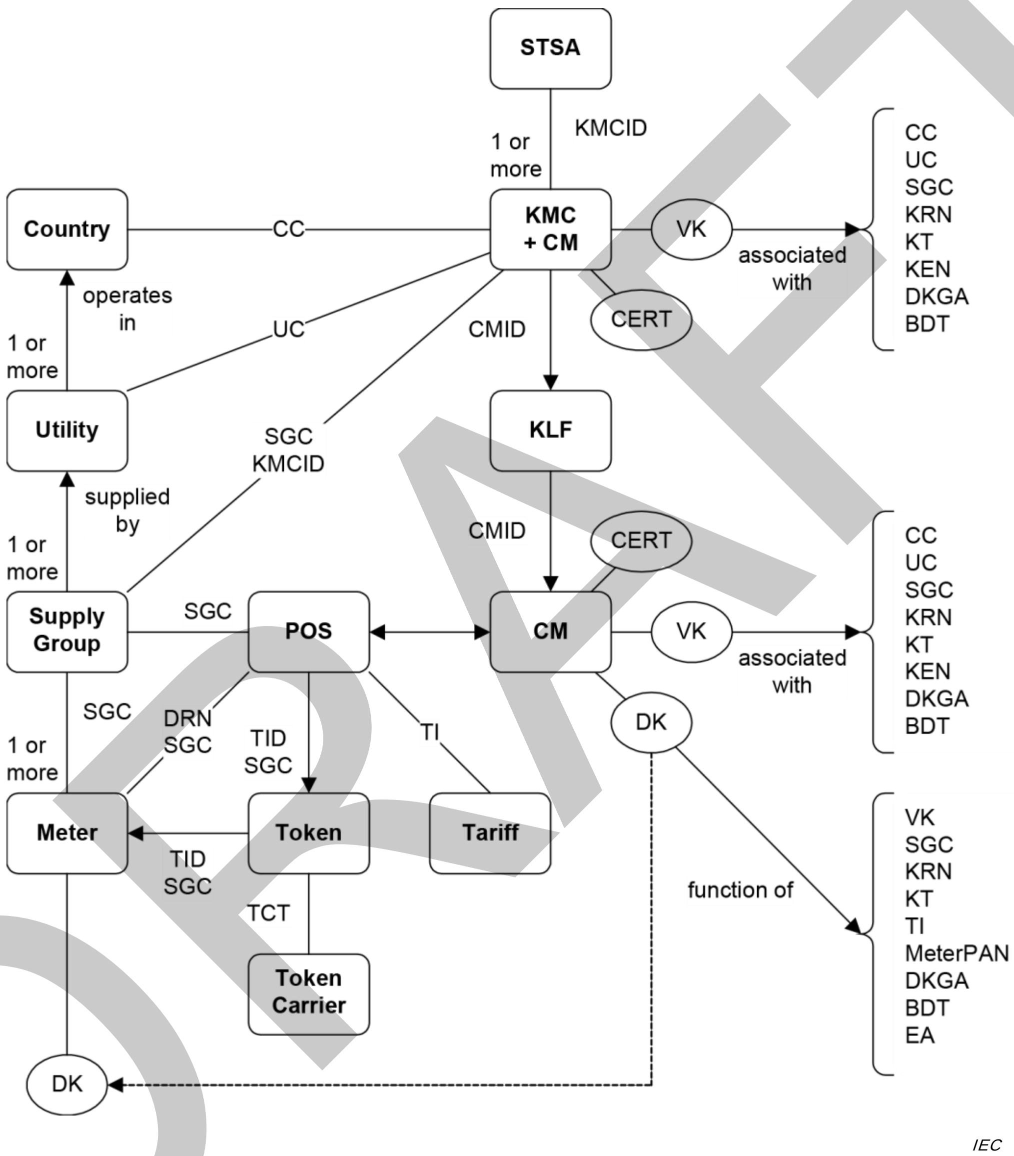


Figure B.1 – Entities and identifiers deployed in an STS-compliant system

For the maintenance of these entities and related services see Clause 10.

The entities that are typically deployed in an STS-compliant system are given in Table B.1.

DRAFT

Table B.1 – Typical entities deployed in an STS-compliant system

Entity	Context	Reference
Country	Geographical area with politically demarcated boundaries, which may change over time	x
Utility	Entity that supplies a service like electricity to its end customer by means of a payment meter. One or more utilities are operational in a country. Utilities change their constitutional identities over time	x
SupplyGroup	A subgroup of payment meters within a distribution network. A Utility may supply to one or more SupplyGroups. A SupplyGroup may change its relationship to a Country and a Utility over time	6.1.6
Meter	The payment meter used to control the delivery or supply of the service to the end customer (see also IEC 62055-31). One or more payment meters are grouped in a SupplyGroup. A payment meter may change to a different SupplyGroup by means of a corresponding DecoderKey change	IEC 62055-31 IEC TR 62055-21
POS	PointOfSale device that is able to generate tokens for any payment meter in a SupplyGroup, by having access to the VendingKey value for the particular SupplyGroup. It is technically and practically feasible that a POS may have access to VendingKey values of more than one SupplyGroup, thus being able to also generate tokens for payment meters belonging to those SupplyGroups. VendingKeys may thus move to and from PointOfSale devices over time, depending on the commercial relationship between a vendor and a particular Utility	IEC TR 62055-21
TokenCarrier	The physical device, or medium onto which the token information is encoded and which is then used to transfer the token to the payment meter. This may be in the form of a printed numeric string or a magnetically encoded card, which is carried to the payment meter by hand and manually inserted into the reading device of the payment meter by the user (end customer), or it may be a virtual token carrier in the form of a direct communication connection to a remotely located client device	3.1
Token	Token as defined in this standard by means of which the POS device is able to transfer instructions and information to the payment meter, or retrieve information from the payment meter	3.1
Tariff	The formula used to calculate the charge per unit of service. In the case of the one-way payment meters the tariff is normally applied at the POS at the time when the end customer purchases a token. There are normally several tariff structures according to different customer categories and contracts. Each tariff is thus associated with a TI (see below) for ease of reference	6.1.7 6.2.6
STSA	Standard Transfer Specification Association that keeps a register of all KMCs, which are globally deployed	Clause C.1
KMC	KeyManagementCentre. The infrastructure that is used to manage and control the KeyManagementSystem. It includes a CM.	Clause 9 Annex A Clause C.3
KLF	KeyLoadFile. The secure mechanism used by the KMC to distribute VendingKey values to cryptographic modules	Annex A
CM	CryptographicModule. The secure device used by the KMC to generate VendingKey values and to securely distribute VendingKey values to a CM device located at a POS The secure device used by the POS to generate DecoderKey values from VendingKey values and to generate tokens from DecoderKey values	Annex A
CERT	Certified public key of the KMC CM and the POS CM, which are used to authenticate each entity and to establish a KEK during VK distribution from the KMC CM to the POS CM	x
VK	VendingKey. A secret key value, generated, stored and distributed by the KMC to other cryptographic modules under controlled and authorised conditions when required. It is used to generate DecoderKey values inside the CM	6.5.2.2

Entity	Context	Reference
DK	DecoderKey. A secret key value generated as a function of several parameter values: DK = f (VK, SGC, KRN, KT, TI, MeterPAN, DKGA, BDT, EA). It is shared between the CM and the payment meter and is used to encrypt and decrypt tokens that are sent from the POS to payment meter or from the payment meter to the POS	6.5.2.3

The identifiers that are associated with the above entities are given in Table B.2.

Table B.2 – Identifiers associated with the entities in an STS-compliant system

Identifier	Context	Reference
CC	CountryCode A code uniquely identifying the country in which the Utility is operative and where the payment meters are installed. It is registered in the KMC and associated with VK at the KMC and the CM	x
UC	UtilityCode A code allocated by the KMC to uniquely identify the specific Utility to which VK and the SGC is allocated. It is registered in the KMC and is associated with VK at the CM	x
KMCID	KeyManagementCentreIdentifier Unique identifier for each KMC in the world. Each KMCID is registered with the STSA	x
CMID	CryptographicModuleIdentifier Unique identifier for each cryptographic module in the system	x
TID	TokenIdentifier Unique time-based identifier for each token. It is shared between the POS, the token and the payment meter	6.3.5.1
MeterPAN	MeterPrimaryAccountNumber A unique identification number for each STS-compliant payment meter. It is shared between the payment meter and the POS. Encoding it into the DecoderKey enforces the association with the payment meter	6.1.2
DRN	DecoderReferenceNumber The unique number as it appears in the MeterPAN. It is shared between the POS and the payment meter	6.1.2.3
TCT	TokenCarrierType The type of medium that is used onto which the token is encoded for transfer to the payment meter	6.1.3
SGC	SupplyGroupCode Unique number allocated by the KMC to identify a SupplyGroup of the Utility. It is shared between the SupplyGroup, the KMC and the POS. It is associated with the VendingKey value and recorded in the KMC and also in the CM. Encoding it into the DecoderKey enforces the association with the payment meter	6.1.6
TI	TariffIndex The index number to a register of tariffs associated with a particular Tariff for each customer. It is shared between the Tariff and the POS. Encoding it into the DecoderKey enforces the association with the payment meter. This means that the DecoderKey shall change if the customer is moved onto a different tariff structure	6.1.7
KRN	KeyRevisionNumber Revision of the VendingKey as allocated by the KMC. It is associated with the VendingKey value at the KMC and at the CM. Encoding it into the DecoderKey enforces the association with the payment meter	6.1.8

DRAFT

Identifier	Context	Reference
KT	KeyType The type of the VendingKey as allocated by the KMC. It is associated with the VendingKey value at the KMC and at the CM. Encoding it into the DecoderKey enforces the association with the payment meter	6.1.9
KEN	KeyExpiryNumber A number that is associated with a validity period for the VendingKey. It is associated with the VendingKey value at the KMC and at the CM. It is not encoded in the DecoderKey, but is transferred to the DecoderKeyRegister by means of the key change tokens	6.1.10

DRAFT

Annex C
(informative)

Code of practice for the implementation of STS-compliant systems

C.1 General

The term "must" is used to indicate requirements only in the context of the code of practice as described in this informative Annex and does not impose normative requirements on this standard.

The term "users of the STS" is defined in 10.1.

C.2 Maintenance and support services provided by the STS Association

The STS Association is a not-for-gain company incorporated in South Africa with members comprising of manufacturers of payment meters and associated vending systems and of utilities. The object of the STS Association is to promote the use of the STS, develop the functionality further and maintain the required infrastructure to provide supporting services like key management, product certification and standardisation to users of the STS.

See also Clause 10 for more details on the maintenance of STS entities and related services.

The General Secretary of the STS Association can be contacted at the address given in the introduction to this document. E-mail is the preferred mechanism for correspondence with the STS Association.

C.3 Key management

C.3.1 Key management services

(See also Annex A.)

The STS Association operates a KMC and provides key management services to utilities and STS-compliant product manufacturers worldwide in accordance with this document.

C.3.2 SupplyGroupCode and VendingKey distribution

C.3.2.1 Data elements associated with a SGC

(See also 6.1.6).

The KMC ensures unique allocation of SGC values in accordance with this document.

The KMC generates, stores and distributes VDDK, VUDK and VCDK values with the associated KRN, KT and KEN in accordance with this document.

The KMC ensures that VendingKey values are available to all manufacturers of STS-certified products in accordance with this document.

In order to effectively manage the generation, storage and distribution of SGC and associated VendingKey values, it is recommended that the data elements given in Table C.1 be recorded and be uniquely associated with an SGC.

DRAFT

Table C.1 – Data elements associated with a SGC

Element	Context	Reference
SGC	Actual value of the SupplyGroupCode as registered in the KMC	6.1.6
Country	CountryCode as the country where the SGC and VendingKey is to be used	Annex B
Location	Place associated with the SupplyGroup demarcation (Country, State, Province, City, Town, Suburb)	x
Network	Network associated with the SupplyGroup demarcation (name, ID)	x
Owner	To whom this SGC is allocated: UtilityCode (if applicable) Name of Organization (utility) Address (postal, physical, website) Contact person and details (name, postal, email, tel, fax) Authorization signatory (name, contact details)	x
OwnerHistory	Record of changes to ownership association of the SGC over time	x
LocationHistory	Record of changes to location association of the SGC over time	x
NetworkHistory	Record of changes to network association of the SGC over time	x
KMC	KMCID and country of origin of the KMC as the source of the SGC and VendingKey	Clause 9 Annex A
VendingKey	VendingKey plus attributes (KRN, KT, KEN). These values are in encrypted format	6.5.2 6.1.8 6.1.9 6.1.10
SGCDistribution Register	Register of SGC v/s CM ID (i.e. to which cryptographic modules a particular SGC has been distributed over time)	x

C.3.2.2 SupplyGroupCode demarcation guidelines

This topic is dealt with comprehensively in the STS Association Code of practice (see Bibliography). For the sake of providing some indicators herein, some factors to be taken into consideration are given below.

Factors to consider in deciding the SGC demarcations:

- security risk in terms of compromising a VendingKey;
- security risk in terms of stolen POS devices;
- logistics for payment meter spares;
- control of POS vending agents in authorizing them to vend to the group;
- logistics for separating collected revenue from POS vending agents;
- particular business logic around distribution network maintenance and supply logistics,
- cross-vending rules on SGC boundaries;
- change of payment meter ownership over time (deregulated markets);
- change of supplier over time (deregulated markets).

C.3.3 CryptographicModule distribution

(See also Annex A).

In order to effectively manage the distribution of SGC and VendingKey values to cryptographic modules, it is recommended that the data elements given in Table C.2 be recorded.

Table C.2 – Data elements associated with the CryptographicModule

Element	Context	Reference
CM	Attributes of the CryptographicModule (CMID, CMType, HardwareVersion, Softwareversion,CERT).	Annex A Annex B
CMManufacturer	Name and contact details of organization	Annex A
CMOwner	To whom this CM belongs: UtilityCode (if applicable) Name of Organization (utility) Address (postal, physical, website) Contact person and details (name, postal, email, tel, fax) Responsible person (name, contact details)	Annex A
CMLocation	Details of intended destination of CM where it is going to be used (country, state, province, city, town, suburb)	x
KMC	KMCID and country of origin which initialised the particular CM	Clause 9 Annex A
CMOwnerHistory	Historical register of ownership changes to cryptographic modules over time	x
CMLocationHistory	Historical register of location changes to cryptographic modules over time	x

C.3.4 Key expiry

(See also 6.1.10, 6.5.2.6, 7.3.1.1).

In the case where key expiry for VendingKeys is not dynamically implemented in an STS-compliant installation, then it is the recommended practice to set the KEN to 255.

At the date of publication of this document the key expiry option for DecoderKeys in payment meters had not been implemented in any STS-compliant installation.

C.4 MeterPAN

C.4.1 General practice

(See also 6.1.2).

The MeterPAN serves to uniquely identify each payment meter in the STS-compliant installation worldwide, thus being able to tag and route transactions accordingly. All users of the STS are thus encouraged to follow this practice, which is in line with that of the banking and financial transaction management (see also ISO 4909).

C.4.2 IssuerIdentificationNumbers

As clarified in 6.1.2.2, the IIN for 2-digit Manufacturer Codes is 600727. For 4-digit Manufacturer Codes the IIN is 0000.

C.4.3 ManufacturerCodes

(See also 6.1.2.3.2).

DRAFT

MfrCode values are allocated and managed by the STS Association to ensure uniqueness of the series globally, thus ensuring uniqueness of the MeterPAN globally. Note that both 2-digit and 4-digit Manufacturer Codes may exist.

The current list of MfrCode values can be obtained from the General Secretary of the STS Association (see Clause C.1 for contact details).

C.4.4 DecoderSerialNumbers

(See also 6.1.2.3.3).

Each MeterManufacturer manages his 8-digit range of numbers as he sees fit, as long as it complies with the requirements of this document.

C.5 SpecialReservedTokenIdentifier

(See also 6.3.5.2).

Each utility is free to determine the rules for how this SpecialReservedTokenIdentifier is to be used as a special application to satisfy his special needs.

An example of using this SpecialReservedTokenIdentifier in a special application is as follows: Each household in an installation may collect a government grant in the form of a free token to the value of 50 kWh per month. Such a token may be collected on any day of the month and as many times as is desired, but the payment meter should only accept the first token of such a type in each month. A solution to this problem is to rule that the SpecialReservedTokenIdentifier is to be used for this token type in this particular installation. Such a token may then be generated at any time during the month, because it will always use the 1st day 00h01 time stamp and the payment meter will only accept the first token so generated and reject any subsequent copies as “Used”.

C.6 Permutation and substitution tables for the STA

The STS Association is registered with the IEC as a Registration Authority to provide maintenance services in support of the IEC 62055-4x and 62055-5x series of standards. As part of this service, the STS Association provides the actual values for the permutation and substitution tables (Table 44, Table 45, Table 51 and Table 52) required in 6.5.4.2, 6.5.4.3, 7.3.3.2 and 7.3.3.3 to users of the standard upon request. The contact details for the STS Association are given in Clause C.1 or may be obtained from the IEC website.

C.7 EA codes

(See also 6.1.5).

As this document evolves there will be more EA codes required. This should take place through the normal route via National Committees to the IEC TC 13 as New Work Item Proposals in liaison with the STS Association.

C.8 TokenCarrierType codes

(See also 6.1.3).

As this document evolves there will be more TCT values required. This should take place through the normal route via National Committees to the IEC TC 13 as New Work Item Proposals in liaison with the STS Association.

DRAFT

C.9 MeterFunctionObject instances / companion specifications

A MeterFunctionObject (MFO) is an object-oriented specification that encapsulates a certain functionality of a payment meter. Each MFO is defined in a companion specification and allocated a unique FunctionObjectIdentificationNumber (FOIN).

The STS Association administers the registration of MFO instances and reserves the exclusive rights to allocate FOIN values in the form of companion specifications.

An MFO instance is proposed to the STS Association as a NWIP, after which it is assigned a unique FOIN. The STS Association then publishes the MFO in the form of a companion specification.

See also STS 200-1 (see Bibliography) for more information on function object classes and STS 201-1 (see Bibliography) for an example of a companion specification.

C.10 TariffIndex

(See also 6.1.7).

The utility has the choice of 2 options:

- link the TI to his list of tariff structures and thus link each customer to a TI. This means the DecoderKey changes if the customer is changed from one tariff structure to another, because the associated TI will change;
- fix the TI to a constant value of say = 01 for the life time duration of the payment meter installation and then link each customer to the list of tariff structures in the management system, independent from the TI. This means that the DecoderKey does not have to change when moving a customer from one tariff structure to another.

At the date of publication of this document, most utilities preferred to follow option 2. The main consideration is that it is a major logistical operation to do a key change to a payment meter that is already installed, so this tends to be avoided where possible.

C.11 STS-compliance certification

C.11.1 IEC certification services

The IEC does not provide certification services for products as such and is thus reliant on outside facilities to do this.

C.11.2 Products

The STS Association provides the service to manufacturers of products to facilitate the testing and will provide STS-certification on the basis of the test results.

C.11.3 Certification authority

In due course the STS Association will be in a position to authorize agents that may provide STS-certification services on its behalf.

C.12 Procurement options for users of STS-compliant systems

This document provides for a variety of options, the details of which need to be specified at the time when products and systems are purchased from manufacturers and suppliers.

DRAFT

As a general guide to purchase orders or tender specifications, the items given in Table C.3 are noted.

Table C.3 – Items that should be noted in purchase orders and tenders

Item	Context	Reference
EA	Which algorithm is be used for token encryption in the vending system and for decryption in payment meter. Options: • STA code 07; • MISTY1 code 11. The purchaser should ensure that the tender specification for the payment meters requires that the payment meter labelling shall include the appropriate EA code	6.1.5
TCT	Which TokenCarrierType the payment meter or the vending system should support. Options: • magnetic card type 01; • numeric type 02; • virtual token carrier code 07; • virtual token carrier code 08.	6.1.3
DKGA	Which algorithm the MeterManufacturer or the vending system should use for generating the DecoderKey; Options: • DEA (DKGA01); only for vending systems serving legacy payment meters; • DEA (DKGA02); current systems; • KDF-HMAC-SHA-256 (DKGA04).	6.1.4
CC	Which destination CountryCode the SGC is to be associated with at the KMC. Options: • one of the standard set of ISO Country Codes	Annex B
UC	Which UtilityCode the SGC is to be associated with at the KMC. Options: • existing UC as allocated by KMC; • new UC as allocated by KMC	Annex B
KMCID	Which KMC is to be used for obtaining the VendingKey and the SGC. The MeterManufacturer and the vending system need the specific VendingKey to generate DecoderKeys. Options: • STSA-KMC-1; STS Association KMC currently in operation; • xxx; future possible KMC of choice or relevance	Annex B

DRAFT

<table>
<tr><th>Item</th><th>Context</th><th>Reference</th></tr>
<tr><td>SGC</td><td>Which SGC should the MeterManufacturer or the vending system use for generating the DecoderKeys?
Options:
• xxxxxx existing SGC; obtained from KMC;
• new SGC; for new projects, apply to KMC.
Which KT is, or should be, associated with this SGC?
Options:
• default; MeterManufacturer key;
• unique; utility key;
• common; utility key</td><td>6.1.6</td></tr>
<tr><td>TI</td><td>Which TariffIndex is to be used by the MeterManufacturer and the vending system to generate DecoderKeys?
Options:
• 00-99; (new);
• 00-99; (existing);
• link TI to the tariff table in the vending system; (NOTE 1);
• do not link TI to the tariff table in the vending system. (NOTE 2).
NOTE 1 When the TI is linked to the tariff table in the vending system database then the consumer may be moved to a different tariff structure only by allocation of another associated TI. This means that that the DecoderKey needs to be changed accordingly.
NOTE 2 When the TI is not linked to the tariff table in the vending system database then the consumer may be moved to a different tariff structure without being allocated to another associated TI. This means that that the DecoderKey does not need to be changed</td><td>6.1.7</td></tr>
<tr><td>KRN</td><td>Which KeyRevisionNumber is to be used by the MeterManufacturer and the vending system to generate DecoderKeys?
This information is associated with the SGC VendingKey and is under the control of the KMC from where it should be obtained</td><td>6.1.8</td></tr>
<tr><td>KT</td><td>Which KT is to be used by the MeterManufacturer and the vending system to generate DecoderKeys?
This information is associated with the SGC VendingKey and is under the control of the KMC from where it should be obtained</td><td>6.1.9</td></tr>
<tr><td>KEN</td><td>Which KeyExpiryNumber is to be used by the MeterManufacturer and the vending system to generate DecoderKeys?
This information is associated with the SGC VendingKey and is under the control of the KMC from where it should be obtained</td><td>6.1.10</td></tr>
<tr><td>DecoderKey expiry</td><td>Whether the DecoderKeys should expire or not, using the KEN.
Options:
• shall not expire (this is the current recommended practice);
• shall expire. (this implies periodic DecoderKey changes)</td><td>6.1.10</td></tr>
<tr><td>Meter dispatching key</td><td>Which DecoderKey type the MeterManufacturer should load into the payment meter.
Options:
• DDTK (manufacturer Default key);
• DUTK (utility Unique key);
• DCTK (utility Common key)</td><td>6.1.6</td></tr>
</table>

DRAFT

Item	Context	Reference
Tokens	Which tokens the payment meter or vending system should support. Options: • TransferCredit; • InitiateMeterTest/Display; • SetMaximumPowerLimit; (optional) • ClearCredit; • SetTariffRate; (currency-based accounting payment meters only) • key change tokens; • ClearTamperCondition; (optional) • SetMaximumPhasePowerUnbalanceLimit; (optional for poly phase) • SetWaterMeterFactor. (water payment meters only)	6.2.1
Vending classification	Which functions the vending systems should support. Options: • vending; (vending of credit tokens) (signified by “V”); • engineering; (vending of management tokens) (signified by “E”); • key change. (vending of key change tokens) (signified by “K”). An STS-compliant vending system may provide any combination of one or all of the options listed. If approved by the STS Association, then the corresponding letters may be displayed on the STS logo	x
Credit transfer	Which types of TransferCredit tokens the payment meters or vending system should support. Options: • electricity; • water; • gas; • time; • electricity currency; • water currency; • gas currency; • time currency.	6.2.2
Test/display options	Which types of test and display tokens the payment meters or vending system should support. Options: A list of mandatory and optional tokens are given in 6.3.8	6.3.8
Power limit	Whether the payment meters should provide power limiting and whether the vending system should provide the relevant tokens. Options: • power limit should be implemented or not; • the power limit setting; • how the payment meter should react when the power limit is reached	6.2.4 6.3.9 8.6
Tariff rate	What the tariff rate values are for the payment meters registered in the vending system database and whether the vending system should support the relevant tokens. Options: • preset by manufacturer; • variable and set with token from vending system; • tariff rate per payment meter	6.2.6 6.3.11

DRAFT

Item	Context	Reference
Tamper detection	Whether the payment meters should provide tamper detection and the vending system should support the relevant tokens. Options: • tamper detection should be implemented; • tamper detection should not be implemented; • payment meter should support display tamper status token; • vending system should support display tamper status token. NOTE 3 Clear tamper token support is mandatory with option 1	6.2.9
Phase power unbalance	Whether the payment meters should provide phase power unbalance limiting and the vending system should provide the relevant tokens. Options: • phase power unbalance limiting should be implemented; • phase power unbalance limiting should not be implemented; • preset by manufacturer; • variable and set with token from vending system; • the phase power unbalance limit value; • how the payment meter should react when the phase power unbalance limit is reached	6.2.10 6.3.10 8.12
Initial credit	What the initial value of the credit register of the payment meters should be when it leaves the manufacturer's premises. Options: • cleared to zero; • preset to initial value; • the initial value	x
Special reserved TID	Whether the vending system should implement any special reserved token identifiers. Options: • special reserved token identifiers should not be implemented; • special reserved token identifiers should be implemented; • specified details of special reserved token identifiers	6.3.5.2
STS Certificate of Compliance	The STS-compliant product supplier should provide a copy of the particular product's STS certificate of compliance as issued by the relevant CertificationAuthority	Clause C.11

C.13 Management of TID roll over

C.13.1 Introduction

The Token Identifier (TID) is a 24 bit field, contained in STS compliant tokens, that identifies the date and time of the token generation. It is used to determine if a token has already been used in a payment meter. The TID represents the minutes elapsed since the start of the BaseDate. The incrementing of the 24 bit field every minute of elapsed time means that at some point in time, the TID value will roll over to a zero value.

All STS prepayment meters will be affected by TID roll over on the 24/11/2024. Any tokens generated after this date and utilizing the 24 bit TID will be rejected by the meters as being old tokens as the TID value embedded in the token will have reset back to 0.

DRAFT

In order to remedy this problem all meters will require key change tokens with the roll over bit set. In addition to this, the BaseDate of 01/01/1993 will be required to be changed to the next BaseDate (see 6.1.12). This process will force the meters to reset the TID stack in the meter to 0, and to avoid previously played tokens from being accepted by the meter due to the TID stack reset, the key change process must introduce a new decoder key into the meter.

A process is therefore required to allow for the management of this change with the least impact to the Utilities, equipment suppliers and end customers.

To allow for easier management of large installed bases it is proposed that the following solution manages the change per meter and not per supply group code (SGC) as some Utilities may have a large installed base under a single SGC.

C.13.2 Overview

C.13.2.1 General

Operators responsible for the management of payment meters must ensure adherence to this procedure by all parties involved.

This Code of Practice defines a process for managing vending keys and decoder keys based on different base dates. The following elements, shown in Figure C.1, have been included:

- Key management centre;
- Cryptographic modules;
- Vending systems;
- Meter data upload files;
- Meter manufacturer;
- Meters.

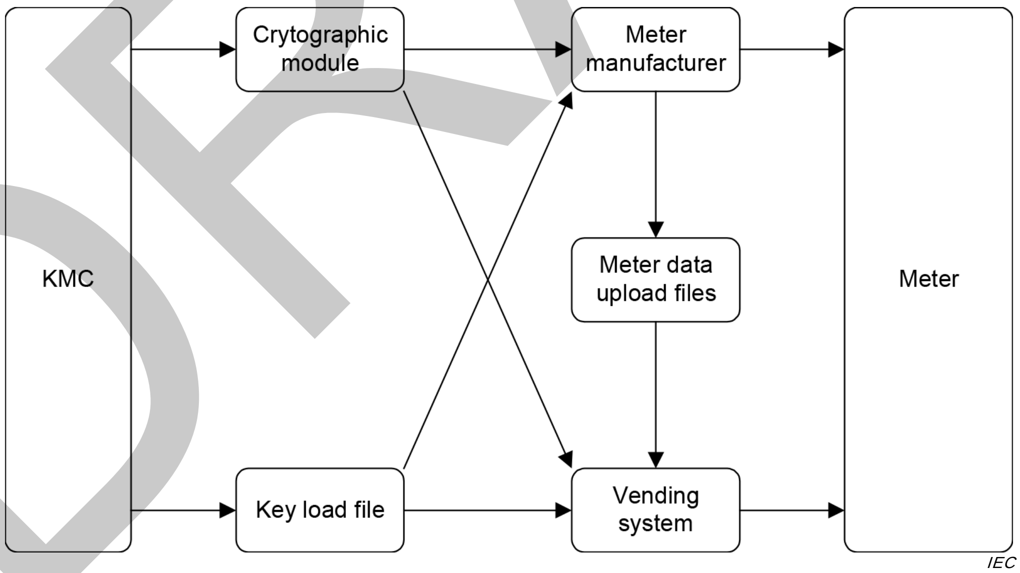


Figure C.1 – System overview

C.13.2.2 Key management centre (KMC)

The KMC is used to generate and load vending keys (VK) into a cryptographic module. The KMC also generates a key load file (KLF) which contains the key load data for a specific cryptographic module to allow a vending system to load VK into the cryptographic module associated with the system. In order to manage the generation of tokens for a specific

DRAFT

BaseDate, the vending system requires the KMC to create a new VK for the new BaseDate interval. The new VK will be created with a different KRN. Associated with each VK in the KLF will be the selected BaseDate. Three BaseDates are supported; namely 01/01/1993, 01/01/2014 and 01/01/2035. It is not envisaged that current technology STS meters will still be in operation by the time the 2035 VK TID rolls over in 2066.

C.13.2.3 Cryptographic module

A cryptographic module will be required to generate key change tokens from a VK on one BaseDate to a VK on a new BaseDate.

C.13.2.4 Vending system

The vending system will be required to manage an associated BaseDate with each VK loaded into a cryptographic module. This BaseDate will be retrieved from the key load file generated at the KMC. Once a new VK is made available, the vending system must allow for the management of the change process whereby a meter or group of meters can be scheduled for a key change. In doing so, the affected meters will undergo a key change with TID roll over thus resetting the meter TID stack and generating a new decoder key based on the new VK. From this point forward all tokens generated for the meter(s) will be encrypted using the new VK with a TID value calculated from the corresponding new BaseDate.

With this process, meters can be scheduled for a key change based on the requirements of the Utility. At any one point in time there may be two or more active vending keys for each SGC as not all meters associated with the SGC will be key changed to the new VK at the same time.

C.13.2.5 Meter upload files

New meters received from the manufacturers can be loaded into the vending system using a meter upload file import process. These meters will be coded by the manufacturers using the latest active VK and therefore each meter record in the meter upload file will be required to include the BaseDate associated with that VK KRN.

C.13.2.6 Meter manufacturers

All meters leaving the factory must be coded using the latest active VK unless otherwise agreed between the utility and the manufacturer. With the three BaseDates chosen, namely 1993, 2014 and 2035, all meters coded before 2014 must be coded using the VK and KRN associated with the BaseDate of 1993. All meters coded between 2014 and 2035 must be coded with the VK and KRN associated with the BaseDate of 2014 and all meters coded after 2035 must utilize the VK and KRN associated with the BaseDate of 2035, unless otherwise agreed between the utility and the manufacturer.

C.13.2.7 Meters

All STS compliant meters must support key change with TID roll over.

C.13.2.8 Key load file

The key load file (KLF) contains the key load data for a specific cryptographic module to allow a vending system to load VK into the cryptographic module associated with that system.

C.13.3 Impact analysis

C.13.3.1 General

The following areas are affected by the above process.

DRAFT

C.13.3.2 Key management centre

- Need to include a BaseDate in the key load file for each VK;
- Support the selection of predefined BaseDates when generating VK;
- Cryptographic Modules must support the key change with TID roll over.

C.13.3.3 Vending systems

- Associate each VK for a SGC with a BaseDate as received from the key load file generated by the KMC;
- May allow meters associated with a previous BaseDate to be scheduled individually, in groups or by SGC for a key change with TID roll over to VK on a new BaseDate.

C.13.3.4 Meter upload files

- Meter Data Upload File specifications must be revised to cater for the addition of the BaseDate.

C.13.3.5 Meter manufacturers

- Must automatically code all meters using the VK with the latest active BaseDate as agreed with the utility;
- Meters must support key change with TID roll over.

C.13.4 Base dates

See 6.3.5 above.

C.13.5 Implementation

C.13.5.1 General

Implementation details for manufacturers of meters and vending systems have been outlined above. The subclauses that follow give basic guidelines for Utilities to follow in the successful implementation of the TID roll over program. Note that Utilities may elect to follow alternative methods of implementation.

C.13.5.2 Assumptions

Prior to starting the implementation of the key-changes in the field, the following are assumed to have been completed by manufacturers of meters, vending systems, and cryptographic modules:

- Secure module firmware has been changed to support the TID roll over functionality;
- Vending software suppliers have modified the vending software to recognise the BaseDates as described in this standard. Once a meter has been key-changed with TID roll over, this event must then be recorded into the vending database;
- All manufactured meters support the TID roll over functionality as specified in IEC 62055-41. Where this is not the case, the meters will have to be changed out with meters that do support the TID roll over functionality. All meters manufactured after the first BaseDate change of 2014, will support the TID roll over functionality.

C.13.5.3 Process for utilities

A guideline to the process to follow is given below:

a) Plan the TID roll over program so as to complete the process at least 1 year before the critical date of 24/11/2024;
b) Communicate the plan, and reasons for the program, to all regions within the utility;

DRAFT

c) Upgrade all vending installations to software and relevant database changes that support the TID roll over functionality;

d) Upgrade utility software to ensure that it supports new Meter Upload file formats, where these are used as an import tool;

e) Upgrade/purchase cryptographic modules with TID roll over functionality through the cryptographic module supplier;

f) Upgrade KMC software to cater for multiple BaseDates;

g) Contact the manufacturer of your meters to confirm whether their meters support key-change with TID roll over. If not, these meters will have to be replaced in the field with meters that do;

h) Start the key-change process.

C.13.5.4 Key-change process

The various following options exist, in no particular order, for the physical execution of the key-change process:

a) Generate key-change tokens for a region and send out technicians to the field to systematically insert these tokens into each meter visited.

b) Generate (automatically) the key-change tokens when a credit purchase is made by the customer. Explain to the customer that the credit token will not function unless the key-change tokens have been entered into the meter first. This is typically the standard practice for key-changes already.

c) Communicate the program to the end customers and request them to collect their key-change tokens by certain deadlines.

All the above options have advantages and disadvantages.

Option a) ensures that the key-changes are done systematically by area, which can then be ‘ticked’ off as completed. This is controllable but expensive in manpower.

Option b) is far less expensive, but does not allow for regions or areas to be done in a controlled fashion since one cannot be sure that tokens have been entered until a new purchase is made. This option also opens the possibility that many complaints will be received regarding non-functional credit tokens if these tokens are entered without the key-change tokens being entered first.

Option c) is the least desirable since communication of the issue goes right to the end customer and may cause unnecessary concerns.

C.13.5.5 Communication of the program

Below is a guideline showing the possible form that the communication to the Utilities regional offices could take. Note that this is a guideline only and may be changed to suit individual utility preferences as required.

"Appropriate addresses and headings.

Subject: Field meter key-change program.

As you may be aware, all prepayment meters store tokens entered as a means to prevent a meter from accepting a token that has already been used. In addition to this storage, each token also has, embedded into the 20 digits, the date and time that the token was generated. The meter then compares this date and time to the oldest token in its memory, and rejects the token if it is older than the oldest token in this memory.

The token date and time field has a maximum range of 31 years. This means that after 31 years of incrementing this date and time field, the value stored will 'roll over' back to zero – much like an odometer in a car going 'round the clock'.

The current tokens will 'roll over' in November 2024 to the current starting date of 1993. At this time, the date and time on the tokens will revert back to its zero date (1993), at which point the meters will no longer accept tokens generated with this base date.

While the date of 2024 may seem like a long time into the future, we need to start making plans to change this base date of 1993 to a later base date. To this end, manufacturers have been made aware that changes will have to be made to the meters, Cryptographic Modules, vending systems, and Key Management Centres to accommodate this change.

The change consists of changing the key in each meter in the field, which can be done by issuing a set of key-change tokens to the customer, or implementing a program whereby each meter is visited by technical staff to enter these tokens.

In order to reduce the number of meters that will have to be visited, or key-changed, in the field, manufacturers will be instructed that all meters made from 2014 onwards, must be coded using the new base date of 2014. This means that the actual number of meters with a base date of 1993 should be dramatically reduced by the time 2024 is upon us, and not many remaining meters will require key-changes.

With the systems currently envisaged by the STS Association, this process should never have to be repeated since the base date of the meters will change every 21 years."

DRAFT

Bibliography

ISO 4217:2015, *Codes for the representation of currencies*

ISO 4909, *Identification cards – Financial transaction cards – Magnetic stripe data content for Track 3*

ISO 16609:2012, *Financial services – Requirements for message authentication using symmetric techniques*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC 9545, *Information technology – Open Systems Interconnection – Application Layer structure*

STS 401-1, *Code of practice for the allocation of supply group codes*

STS 200-1, *Standard transfer specification (STS) – Companion specification – Generic classes for meter function objects*

STS 201-1, *Standard transfer specification (STS) – Companion specification – Meter function object: RegisterTable for electricity payment meters*

STS 402-1, *Code of practice – Management of Token ID Rollover*

STS 600-4-2, *Standard Transfer Specification – Companion Specification – Key Management System*

FIPS PUB 198, *The Keyed-Hash Message Authentication Code (HMAC)*

FIPS PUB 197, *Advanced Encryption Standard*

FIPS PUB 186-2, *Digital Signature Standard*

FIPS PUB 185, *Escrowed Encryption Standard (EES)*

FIPS PUB 180-2, *Secure Hash Standard*

FIPS PUB 171, *Key management using ANSI X9.17*

FIPS PUB 140-2, *Security requirements for cryptographic modules*

FIPS PUB 140-2 Annex A, *Approved security functions for FIPS PUB 140-2, Security requirements for cryptographic modules*

FIPS PUB 140-2 Annex B, *Approved protection profiles for FIPS PUB 140-2, Security requirements for cryptographic modules*

FIPS PUB 140-2 Annex C, *Approved random number generators for FIPS PUB 140-2, Security requirements for cryptographic modules*

FIPS PUB 140-2 Annex D, *Approved key establishment techniques for FIPS PUB 140-2, Security requirements for cryptographic modules*

DRAFT

FIPS PUB 113, *Computer Data Authentication*

FIPS PUB 112, *Password usage*

FIPS PUB 87, *Guidelines for ADP contingency planning*

FIPS PUB 81, *DES modes of operation*

FIPS PUB 74, *Guidelines for implementing and using the NBS Data Encryption Standard*

FIPS PUB 73, *Guidelines for security of computer applications*

FIPS PUB 39, *Glossary for computer systems security*

FIPS PUB 31, *Guidelines to ADP physical security and risk management*

NIST Special Publication 800-38C, *Recommendation for block cipher modes of operation: The CCM mode for Authentication and Confidentiality*

NIST Special Publication 800-38A, *Recommendation for block cipher modes of operation, methods and techniques*

NIST Special Publication 800-20, *Modes of operation validation system for the Triple Data Encryption Algorithm (TMOVS): Requirements and procedures*

NIST Special Publication 800-2, *Public Key Cryptography*

NIST, *NIST-recommended random number generator based on ANSI X9.31 Appendix A.2.4 using the 3-key Triple DES and AES algorithms*

NIST, National Institute for Standards and Technology, *AES key wrap specification*

ANSI X9.62, *Public key cryptography for the financial services industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*

ANSI X9.52, *Triple Data Encryption Algorithm modes of operation*

ANSI X9.42, *Agreement of symmetrical keys on using Diffie-Hellman and MQV algorithms*

ANSI X9.24 Part 1, *Retail Financial Services Symmetric Key Management Part 1: Using Symmetric Techniques*

ANSI X9.31, *Digital signatures using reversible public key cryptography for the financial services industry (rDSA)*

ANSI X9.17, *Financial institution key management (wholesale)*

ANSI X9.9, *Financial institution Message Authentication (wholesale)*

RFC 4868, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*

RFC 4231, *HMAC-SHA Identifiers and Test Vectors December 2005*

FIPS PUB 198-1, *The Keyed-Hash Message Authentication Code (HMAC)*

DRAFT

FIPS PUB 180-1, *Secure Hash Standard (SHS)*

NIST Special Publication 800-108, *Recommendation for Key Derivation Using Pseudorandom Functions*

DRAFT