IEC 62055-41

Edition 2.0 2014-05

INTERNATIONAL STANDARD

NORME INTERNATIONALE

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

Comptage de l'électricité – Systèmes de paiement –
Partie 41: Spécification de transfert normalisé (STS) – Protocole de couche application pour les systèmes de supports de jeton unidirectionnel

IEC 62055-41:2014-05(en-fr)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue
The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de l'IEC
La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC
Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue
Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub
La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,…). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished
Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org
Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary
Plus de 55 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc
Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

IEC 62055-41

Edition 2.0 2014-05

INTERNATIONAL STANDARD

NORME INTERNATIONALE

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

Comptage de l'électricité – Systèmes de paiement – Partie 41: Spécification de transfert normalisé (STS) – Protocole de couche application pour les systèmes de supports de jeton unidirectionnel

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

PRICE CODE
CODE PRIX XE

ICS 17.220.20; 35.100.70; 91.140.50

ISBN 978-2-8322-1487-9

CONTENTS

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.

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

This second edition cancels and replaces the first edition issued in 2007. It constitutes a technical revision. The main technical changes with regard to the previous edition are as follows:

- Class 2 token is extended to include credit transfer for gas and water with associated extensions in the display/test tokens.
- MfrCode is extended from 2 to 4 digits.
- Three token identifier base dates are defined to provide for more frequent key changes with TID roll-over procedures.
- A code of practice for the management of TID roll-over key changes in association with the revised set of base dates.
- Some clarifications and additional examples have been added.

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

CDV	Report on voting
13/1530/CDV	13/1553/RVC

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

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.

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

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

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

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 standard, 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 – Application layer protocol for one-way token carrier systems

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

Part 52: Standard transfer specification – 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.

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 standard are compulsory for implementation in a particular system configuration and as a guideline, a selection of optional configuration parameters are listed in Clause C.11.

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 IEC 62055-41 Ed 1 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 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 130 organisations located in over 24 countries. Interoperability and conformance to the Standard Transfer System (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 35 million meters operated by 400 utilities in 30 countries. Management of the technology has been administered by the STS Association in fulfilment of its role as the IEC appointed Registration Authority.

Global success has brought about an urgent need to extend the range of the numerical elements contained in IEC 62055-41 tables. In particular, the range of manufacturer numbers need to be extended beyond the 99 numbers originally provided. Also, application of the standard has been extended to cater for multi-energy systems including gas and water meters. Accordingly, there is a need to ensure that the content of IEC 62055-41 is maintained to cater for this market growth and multi-energy extensions.

Several corrections and clarifications are also required to bring Ed 1 up to date with current practice. This was considered by TC13 WG15 at its meeting on the 20 September 2012 in London, where it was agreed that IEC 62055-41 should be revised.

Only the most urgently required revisions have been incorporated in Edition 2 due to timing constraints, but it is anticipated that Edition 3 will consider further revisions to incorporate the following functionalities:

- Currency transfer
- Enhanced security on par with contemporary industry practice
- Complex functions fully harmonized with DLMS/COSEM suite
- Decentralized key management system with distributed architecture
- Conformance certification test suite in conjunction with IECEE CB scheme

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 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: sts@vdw.co.za

Website: http://www.sts.org.za

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 International Standard only, which is the subject of the purchase contract (see also Clause C.11).

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

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60050 (all parts), *International Electrotechnical Vocabulary* (available at <http://www.electropedia.org>)

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

IEC 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 7812-1:2006, *Identification cards – Identification of issuers – Part 1: Numbering system*

ISO/IEC 7812-2:2007, *Identification cards – Identification of issuers – Part 2: Application and registration procedures*

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*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 60050, IEC 62051, IEC 62055-31 as well as the following apply.

NOTE Where there is a difference between the definitions in this standard and those contained in other referenced IEC standards, then those defined in this standard 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 a multi-part payment meter.

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 service provider party.

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.8)

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-part payment meter the DRN and meter serial number may be synonymous, while in a multi-part 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
the ability to perform a defined function

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

3.2 Abbreviations

ANSI	American National Standards Institute
APDU	ApplicationProtocolDataUnit
CA	CertificationAuthority
CC	CountryCode
CIS	Customer Information System
CM	CryptographicModule
CMAC	CryptographicModuleAuthenticationCode
CMID	CryptographicModuleIdentifier
COP	Code of practice
CRC	CyclicRedundancyCode
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
DOE	DateOfExpiry
DRN	DecoderReferenceNumber [known as a “meter number” in systems in use prior to the development of this standard]
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
IAIN	IndividualAccountIdentificationNumber
ID	Identification; Identifier
IIN	IssuerIdentificationNumber
ISDN	Integrated Services Digital Network
ISO	International Standards Organisation
ISO BIN	Replaced by IIN
KCT	KeyChangeToken
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	Rollover
SG	SupplyGroup
SGC	SupplyGroupCode
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
TI	TariffIndex
TID	TokenIdentifier
UC	UtilityCode
VCDK	VendingCommonDESKey
VDDK	VendingDefaultDESKey
VK	VendingKey
VUDK	VendingUniqueDESKey
WAN	Wide Area Network
XOR	Exclusive Or (logical)

3.3 Notation and terminology

Throughout this standard 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 standard, 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".

5 Reference model for the standard transfer specification

5.1 Generic payment meter functional reference diagram

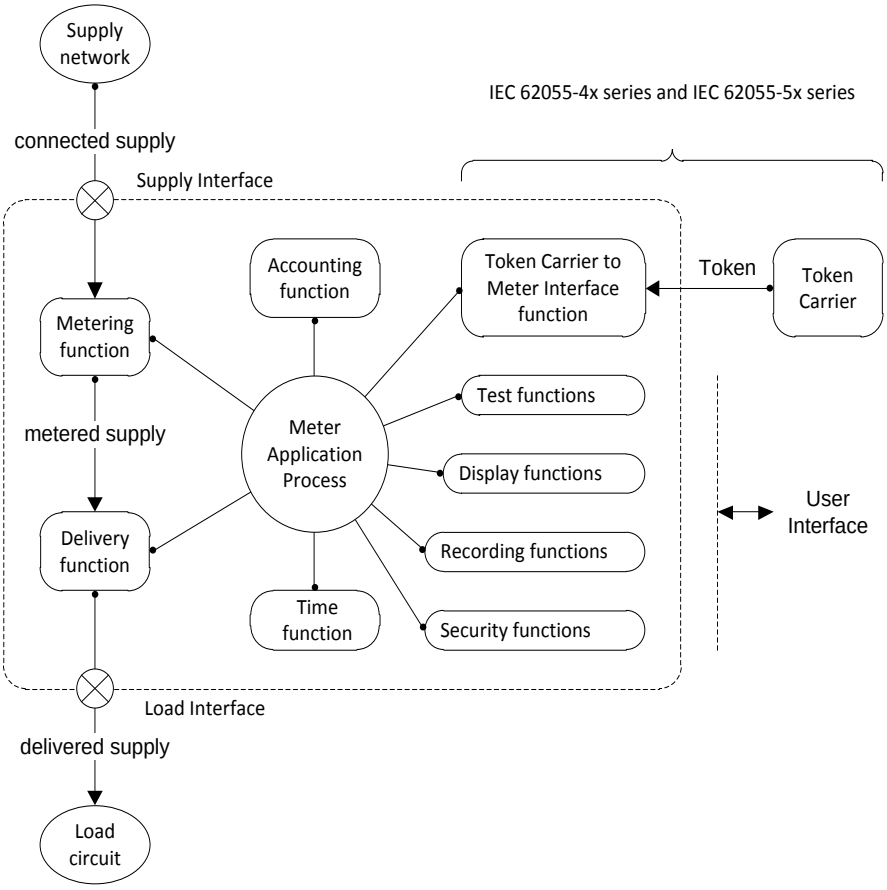


Figure 1 – Functional block diagram of a generic single-part payment meter

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

NOTE MeterFunctionObjects are further discussed in 5.5.

In a single-part payment meter all the essential functions are located in a single enclosure as depicted in Figure 1 above, in which case the decoder is integral with the metering function and the DRN could thus optionally be synonymous with the meter serial number.

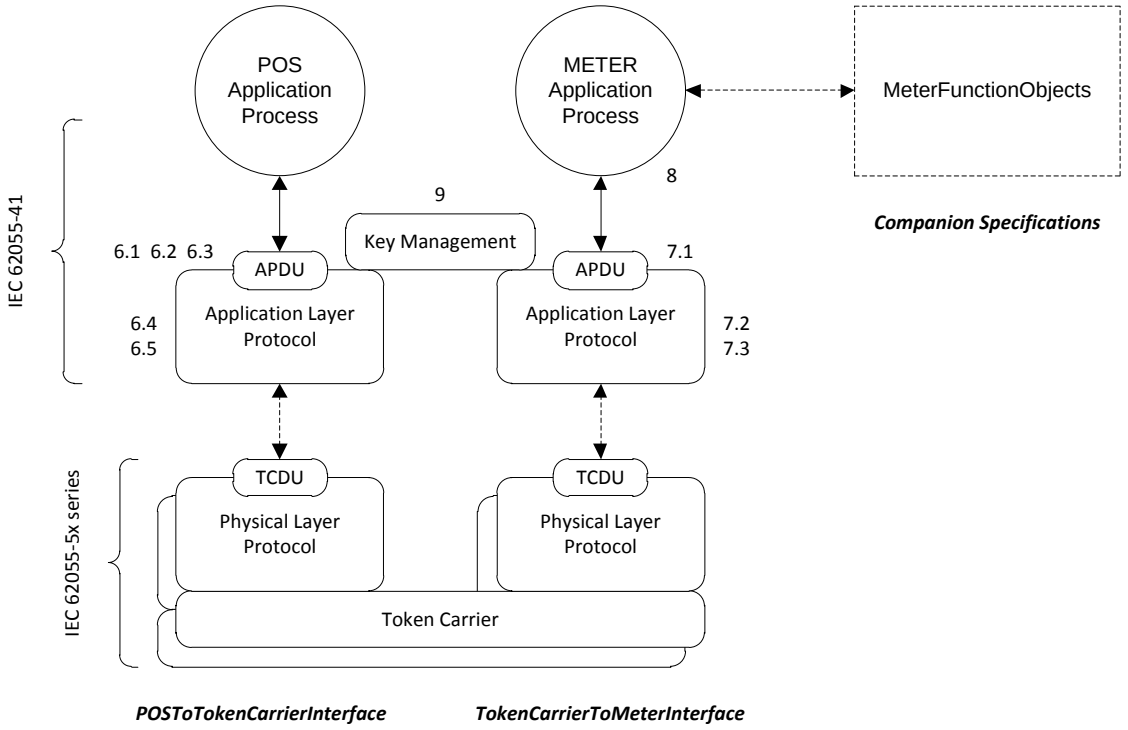
In a multi-part payment meter it is possible for the TokenCarrierToMeterInterface to be located in a separate enclosure from that of the metering function for example, which may well be a standalone meter in its own right and having its own meter serial number. In this case, the DRN would not be the same as the meter serial number, but would be distinctly different and would thus be marked on the enclosure containing the decoder.

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

It is also possible that the Application Layer functions and the Physical Layer functions are located in separate enclosures, in which case, the marking (see 8.3) of the DRN and the EA code is applied to that part that contains the physical TokenCarrier connection point. This may be a cable or modem connector for a virtual token carrier, a keypad for a numeric token carrier or a magnetic card reader for magnetic card token carrier for example (see also 5.2 for more examples of token carriers).

For a more complete description of payment meter function classes see IEC 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 clause number references in this standard 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 standard, 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 standard 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 standard.

5.3 Dataflow from the POSApplicationProcess to the TokenCarrier

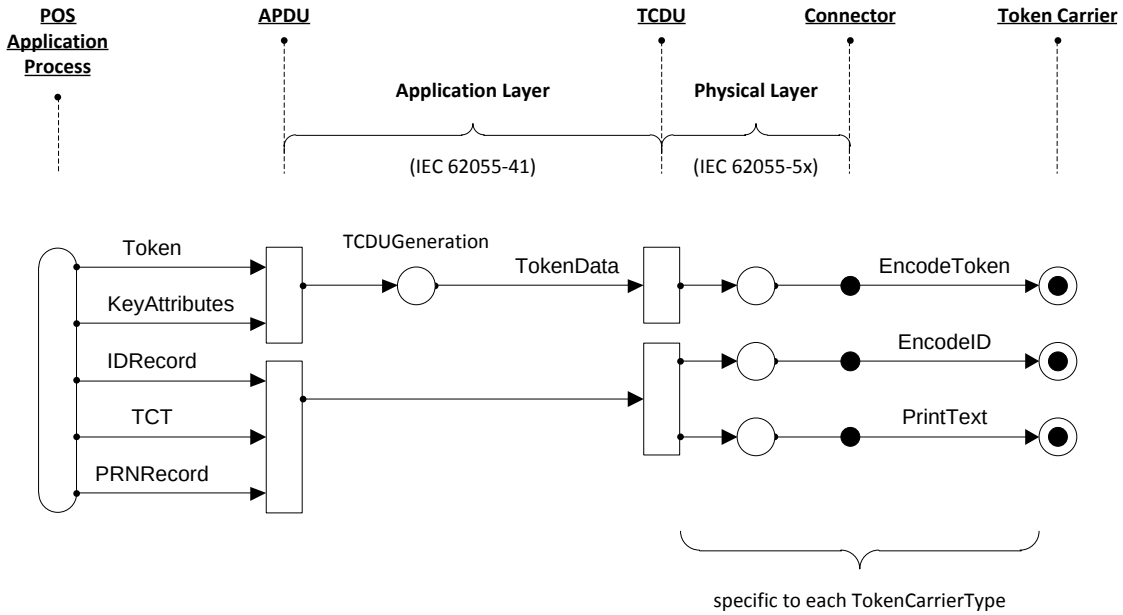


Figure 3 – Dataflow from the POSApplicationProcess to the TokenCarrier

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

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

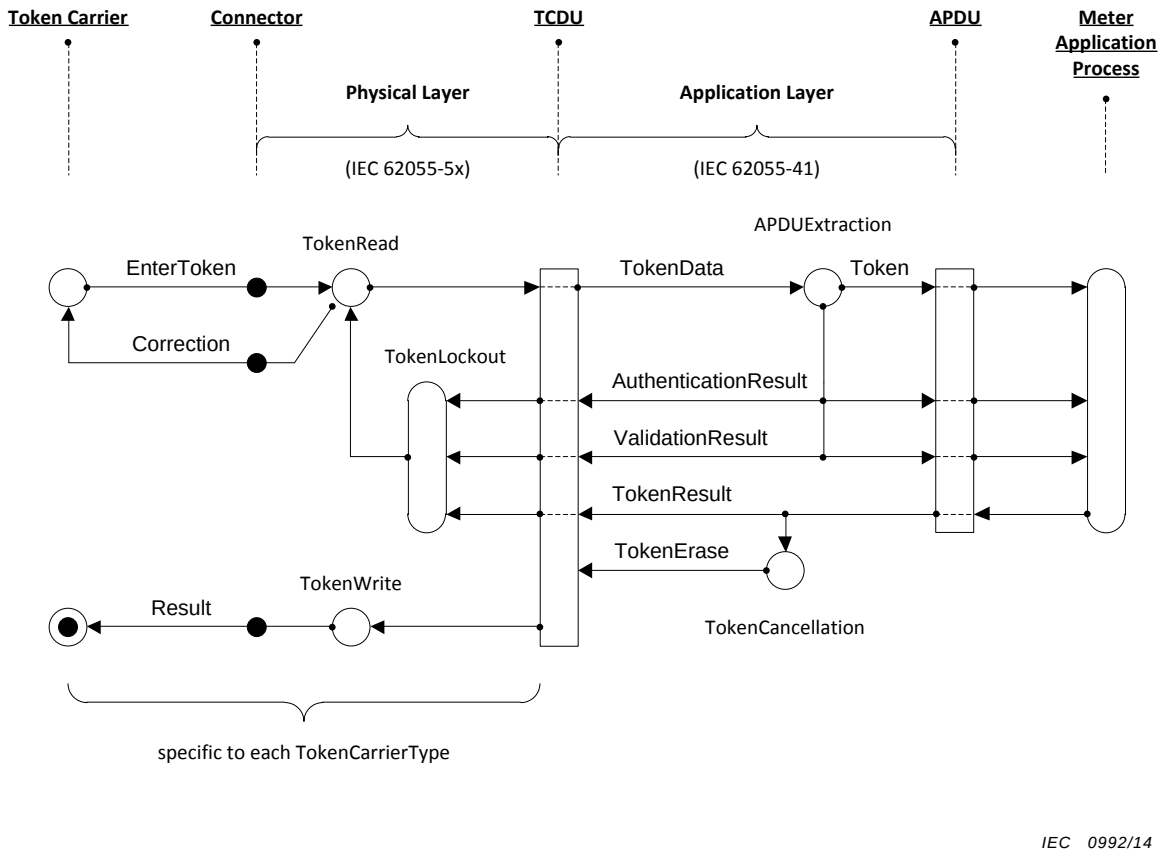


Figure 4 – Dataflow from the TokenCarrier to the MeterApplicationProcess

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

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.

NOTE Other types of connectors would be 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. Such token carriers might 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).

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 standard.

Companion specifications (see Figure 2) are under the administrative control (see Clause C.8) 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 ISO transaction reference numbers

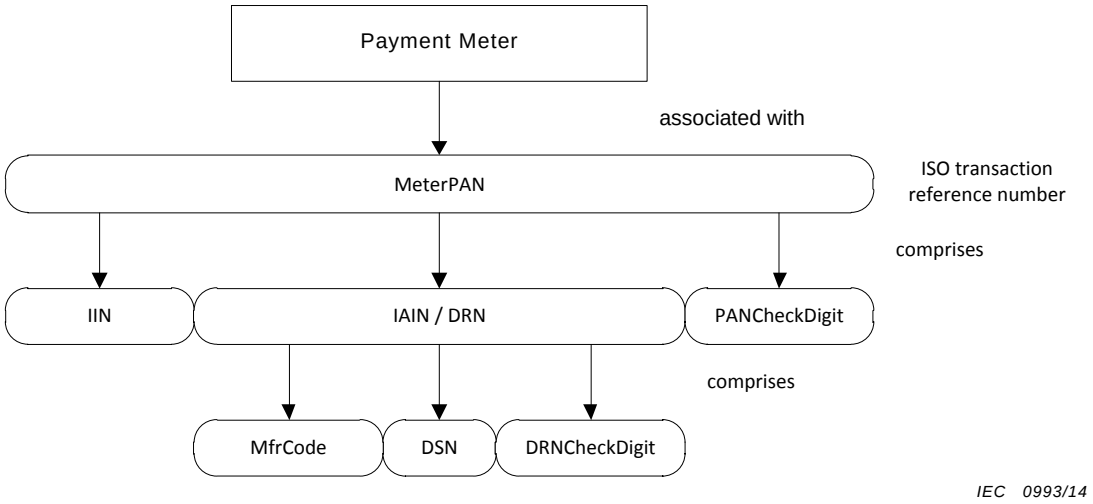


Figure 5 – Composition of ISO transaction reference number

The ISO 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) as defined by ISO/IEC 7812-1 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	ISO compliant 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	1 digit	6.1.8
KT	Directs which KeyType the DecoderKey is on	1 digit	6.1.9
KeyExpiryNumber	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
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	ISO compliant 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	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 and is in accordance with the definition for the PAN (PrimaryAccountNumber) of ISO/IEC 7812-1.

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	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-digit number that defines a domain, under which further IAIN values (i.e. DRN values) may be issued for use within this defined domain.

The original intent and purpose of the IIN was to be able to tag financial transactions in order to uniquely identify them and to route them to the appropriate transacting financial accounts. It was thus intended that the IIN be issued by the ISO under the registration scheme given in ISO/IEC 7812-1 and ISO/IEC 7812-2. However, this has proven to be impractical and the value 600727 for IIN has since become the de-facto standard for legacy systems utilising an 11-digit DRN.

It has subsequently become necessary to also make provision for 13-digit DRNs (as defined in 6.1.2.3.1) in which case the IIN shall be 0000 (four zeroes).

See also C.3.2 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.

This 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's.

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.

See also C.3.3 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.3.4 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 7812-1:2000. 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 7812-1:2000 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
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
08-99	Reserved	For future assignment

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	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 diversification	6.5.3.4
03	DKGA03	System using dual 64-bit DES VendingKey diversification	6.5.3.5
04-99	Reserved	For future assignment	x
DKGA02 is the algorithm to be used for current systems, subject to the criteria for DKGA01. DKGA03 is the algorithm to be used for future systems requiring a higher level of security regarding protection of the VendingKey by “brute-force” attack. Introduction of DKGA03 should preferably coincide with the change from STA to DEA (EA code 07 to EA code 09). 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	x
01-06	Reserved	Legacy proprietary systems	x
07	STA	Systems using the Standard Transfer Algorithm as defined in this standard	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
11-99	Reserved	For future assignment	x
It is recommended that the choice of EA code 09 be co-ordinated with the choice of DKGA03 in order to minimize the effect on existing systems in the installed base (see 6.1.4).			

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 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 a VendingKey associated with it and hence each payment meter in the SupplyGroup has a derived DecoderKey associated with it. 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.

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.2.2 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 of 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 6.2.8) 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.9 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 is associated with a version of the VendingKey and with the corresponding DecoderKey.

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.2.4 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 – 99	2001 – 2099

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

6.2 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 11.

Table 11 –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 Reserved: 3 = time 4 = currency 5-15 = future assignment				
NOTE The SubClass values 3-4 are reserved by the STS Association for applications other than electricity, gas and water, and values 5-15 are reserved for future assignment.					

Action: Transfer credit to the payment meter to the value as defined in the Amount field 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.	Reserved for future assignment.	Reserved for future assignment.	
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.

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.

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) 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				

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

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

6.2.7 Class 2: Set1stSectionDecoderKey

Class	SubClass	KENHO	KRN	RO	Res	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	x	0-3		

Action: Load the DecoderKeyRegister with the 1st half of the new DecoderKey, subject to an authentic loading of a Set2ndSectionDecoderKey token.

6.2.8 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, subject to an authentic loading of a Set1stSectionDecoderKey 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.

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 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.

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	8-10				

Action: Reserved for future definition by the STS Association.

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

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 standard 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
2 bits	4 bits	60 bits
3	0-15	

Action: Reserved for future definition by the STS Association.

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

6.3 Token data elements

6.3.1 Data elements used in tokens

The data elements given in Table 12 are used in tokens in various combinations.

Table 12 – Data elements used in tokens

Element	Name	Format	Reference
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	CyclicRedundancyCode (see 6.2.2 to 6.2.13)	16 bits	6.3.7
KENHO	KeyExpiryNumberHighOrder (see also 6.2.7)	4 bits	6.3.16
KENLO	KeyExpiryNumberLowOrder (see also 6.2.8)	4 bits	6.3.17
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 6.2.8)	32 bits	6.3.15
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	Reserved for future assignment (see also 6.2.7and 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	6.3.18
SubClass	TokenSubClass (see also 6.2.2 to 6.2.14)	4 bits	6.3.3
TI	TariffIndex (see also 6.2.8)	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 13.

Table 13 – Token classes

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

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.

6.3.3 SubClass: TokenSubClass

Further sub-classification of the TokenClass is given in Table 14.

Table 14 – Token sub-classes

<table>
<tr><th rowspan="2">Token SubClass</th><th colspan="4">TokenClass</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 by STS Association for future assignment</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 by STS Association for future assignment</td><td>SetTariffRate</td></tr>
<tr><td>3</td><td>TransferCredit (time)
Reserved by STS Association for connection time applications</td><td>Set1stSectionDecoderKey</td></tr>
<tr><td>4</td><td>TransferCredit (currency)
Reserved by STS Association for currency applications</td><td>Set2ndSectionDecoderKey</td></tr>
<tr><td>5</td><td rowspan="11">Reserved by STS Association for future assignment</td><td>ClearTamperCondition</td></tr>
<tr><td>6</td><td rowspan="3">Reserved for proprietary use for 4-digit MfrCode</td><td>SetMaximumPhasePowerUnbalanceLimit</td></tr>
<tr><td>7</td><td>SetWaterMeterFactor
Reserved by STS Association for water applications</td></tr>
<tr><td>8</td><td rowspan="3">Reserved by STS Association for future assignment</td></tr>
<tr><td>9</td><td rowspan="3">Reserved by STS Association for future assignment</td></tr>
<tr><td>10</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.

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 an STS base date and time. This field is a 24-bit binary representation of the elapsed minutes.

In order to accomodate the fact that the TID will roll over every 31 years, three STS base dates are defined. These are:

- 01 January 1993, 00:00:00;
- 01 January 2014, 00:00:00;
- 01 January 2035, 00:00:00.

With a date and time format of YYYY:MM:DD:hh:mm:ss the STS base date and time of 1993:01:01:00:00:00 corresponds to a TID 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 15.

Table 15 – TID calculation examples

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

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.4 Code of practice for the management of this special reserved TID.

NOTE The use of special application tokens are optional (see Clause C.11), 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

The associated unit for the transfer amount is defined in Table 16.

Table 16 – Units of measure for electricity

Transfer type	Units of measure
Electrical energy	Watt-hours × 100 (0,1 kWh)
Electrical power	Watts

The STS Association also reserves the transfer types given in Table 17 for other applications.

Table 17 – Units of measure for other applications

Transfer type	Units of measure
Water	Litres × 100
Gas	Cubic metres
Time	Minutes
Currency	Under review
NOTE The STS Association defines other future transfer types for other utility services.	

The 16 bits of the transfer 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 18.

Table 18 – Bit allocations for the TransferAmount

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 transfer amount 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 transfer amount;

e is the base 10 exponent;

m is the mantissa; and

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

All transfer amount conversions shall be rounded up in favour of the customer. The possible transfer amount ranges and the associated maximum errors that can arise owing to rounding up are shown in Table 19. Examples of TransferAmount values are given in Table 20.

Table 19 – Maximum error due to rounding

Exponent value	Transfer amount range	Maximum error
0	0000000 to 00016383	0,000
1	0016384 to 00180214	0,061 %
2	0180224 to 01818524	0,055 %
3	1818624 to 18201624	0,055 %

Table 20 – Examples of TransferAmount values for credit transfer

Item	Units purchased	Resultant 16-bit transfer amount field	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

6.3.7 CRC: CyclicRedundancyCode

The CRC is a checksum field used to verify the integrity of the data transferred. 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 21).

Table 21 – 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

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 22.

Table 22 – Permissible control field values

LS Bit No. = 1	Test No	Action	Condition
All bits = 1	0	Do test No. 2 to 5 plus, optionally, any other	Mandatory
1	1	Test the load switch	Optional
2	2	Test the payment meter information display devices	Mandatory
3	3	Display cumulative kWh energy register totals	Mandatory
4	4	Display the KRN	Mandatory
5	5	Display the TI	Mandatory
6	6	Test the token reader device	Optional
7	7	Display maximum power limit	Optional
8	8	Display tamper status	Optional
9	9	Display power consumption	Optional
10	10	Display software version	Optional
11	11	Display phase power unbalance limit	Optional
12	12	Display water meter factor	Mandatory for water payment meter
13	13	Display tariff rate	Mandatory for currency-based payment meter
14-28/36	Reserved	Reserved by STS Association for future assignment	Reserved
NOTE The cumulative kWh energy register is defined in 5.11.4 of IEC 62055-31:2005.			

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.

This option is subject to the supply agreement between the supplier and the utility and shall not form a normative part of this standard.

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 not form a normative part of this standard.

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 by the STS Association for future definition.

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 23.

Table 23 – 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 Currency Credit register
5 to FFFE hex	Reserved for future assignment
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 6.4.5) and which is to be transferred to the payment meter by means of the token.

6.3.16 KENHO: KeyExpiryNumberHighOrder

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

6.3.17 KENLO: KeyExpiryNumberLowOrder

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

6.3.18 RO: RolloverKeyChange

If the RolloverKeyChange bit is set = 1, the payment meter shall perform a rollover 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).

6.4 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.

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.

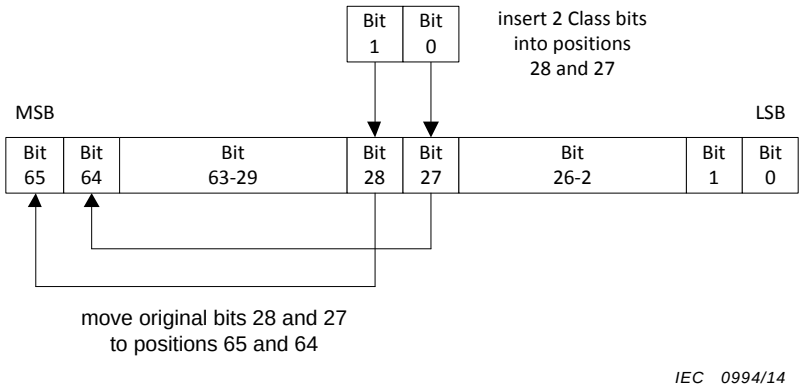


Figure 6 – Transposition of the 2 Class bits

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

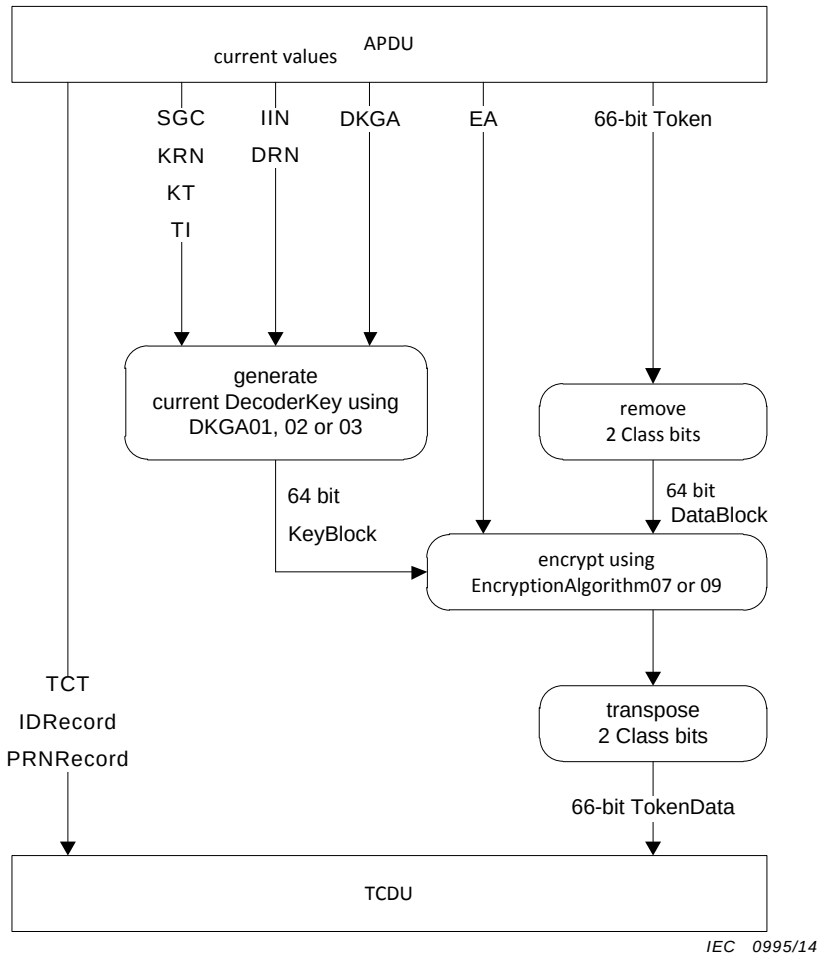


Figure 7 – TCDUGeneration function for Class 0, 1 and 2 tokens

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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens (see 6.4.4 and 6.4.5).

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.

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 and DRN 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 Set1stSectionDecoderKey token

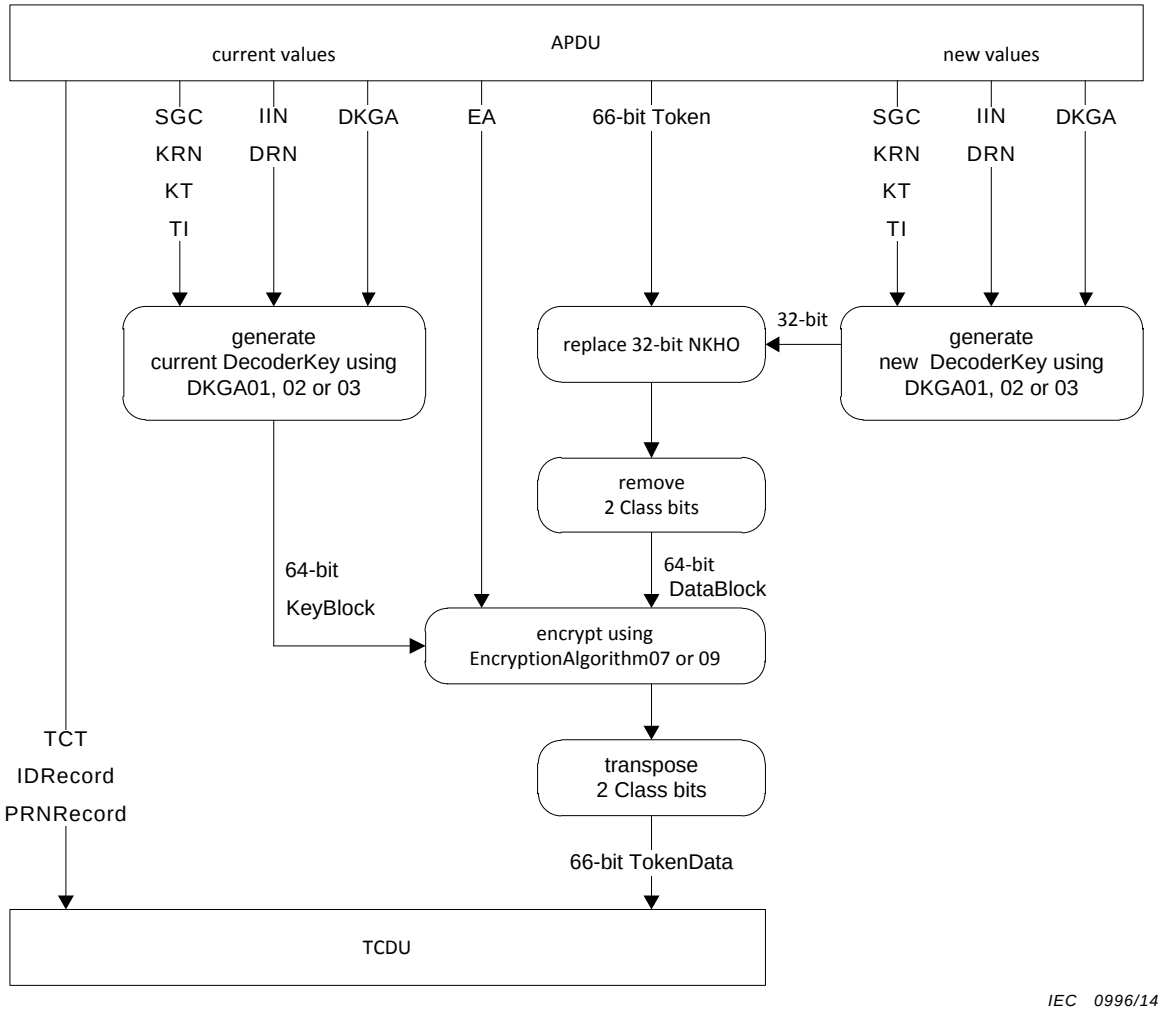


Figure 8 – TCDUGeneration function for Set1stSectionDecoderKey token

This is the transfer function from the APDU to the TCDU (see Figure 8) and is applicable only to the Set1stSectionDecoderKey token.

The Set1stSectionDecoderKey TCDUGeneration function is shown here as being separate from the Set2ndSectionDecoderKey TCDUGeneration function, but in practice the two may be merged into one in order to save on processing resource and for the sake of convenience. In

such a case, the new DecoderKey generation only needs to happen once for example, although the final result is still the same. Thus two separate TCDU instances are always produced: one for the Set1stSectionDecoderKey token and a second for the Set2ndSectionDecoderKey token.

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 and DRN. The specific algorithm to use is in accordance with the value of DKGA in the APDU;
- the resultant new DecoderKey value high order 32 bits are then used to replace the NKHO field of the Set1stSectionDecoderKey token (see 6.2.7) 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 and DRN 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

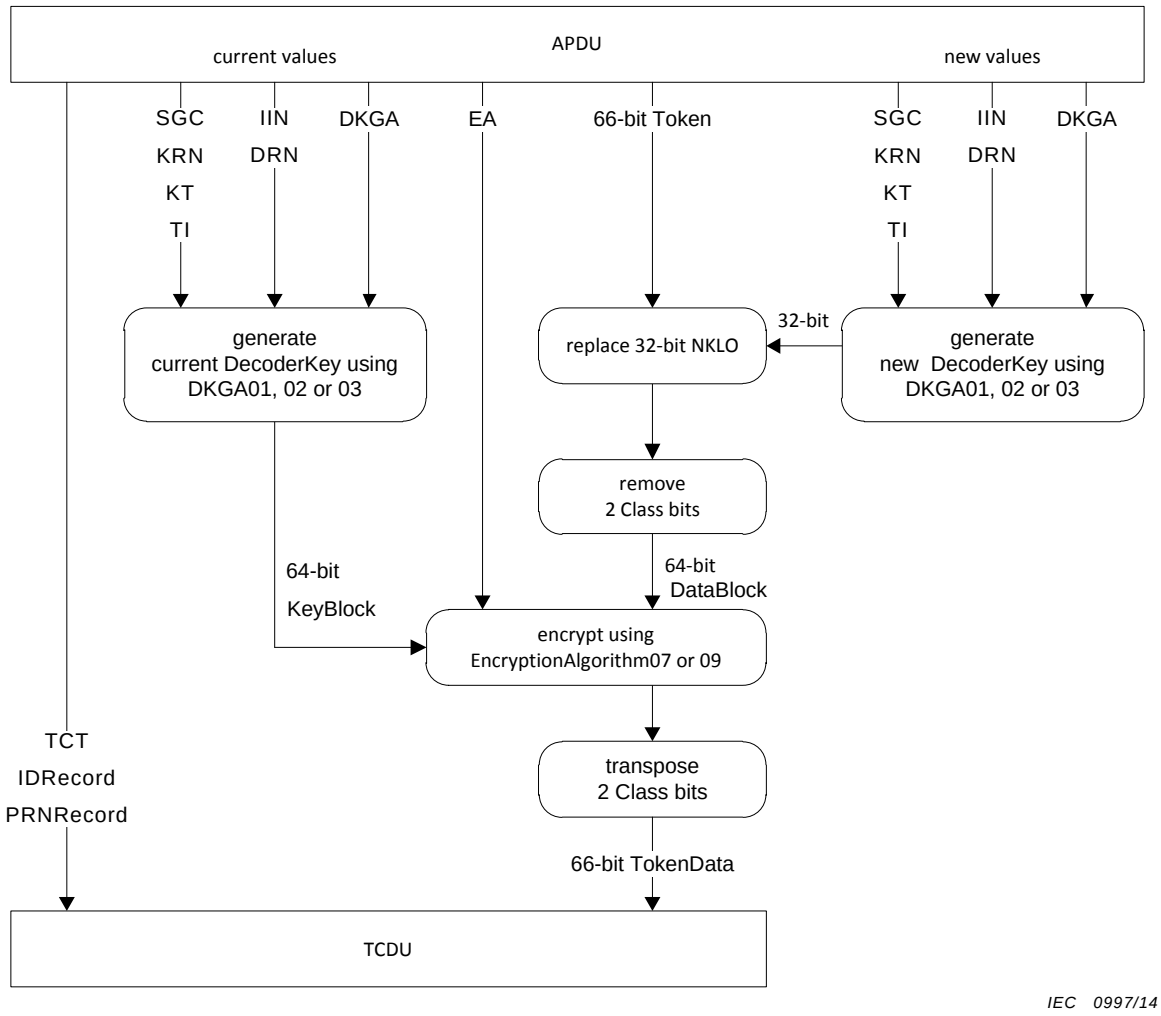


Figure 9 – TCDUGeneration function for Set2ndSectionDecoderKey token

This is the transfer function from the APDU to the TCDU (see Figure 9) and is applicable only to the Set2ndSectionDecoderKey token.

The Set2ndSectionDecoderKey TCDUGeneration function is shown here as being separate from the Set1stSectionDecoderKey TCDUGeneration function, but in practice the two may be merged into one in order to save on processing resource and for the sake of convenience. In such a case, the new DecoderKey generation only needs to happen once for example, although the final result is still the same. Thus two separate TCDU instances are always produced: one for the Set1stSectionDecoderKey token and a second for the Set2ndSectionDecoderKey token.

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 and DRN. The specific decoder key generation algorithm to use is in accordance with the value of DKGA in the APDU;
- the resultant new DecoderKey value low order 32 bits are then used to replace the NKLO field of the Set2ndSectionDecoderKey token (see 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 and DRN 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.

6.5 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey token pair effects an STS key change. This meter-specific management token pair 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 TI.

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

- the value of the VendingKey;
- 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.

NOTE See 6.1.1 for detailed specifications on the data elements in the APDU.

6.5.2.2 VendingKey classification

6.5.2.2.1 Classification of vending keys

The VendingKey is a DES key value that is secretly generated, stored and distributed within the KeyManagementSystem (see Annex A). DES 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.

STS VendingKeys are classified according to the KT values given in Table 24.

Table 24 – Classification of vending keys

KT	SGC type	VendingKey type	Context
0	Initialization	Not specified	Not applicable
1	Default	VDDK	VendingDefaultDESKey
2	Unique	VUDK	VendingUniqueDESKey
3	Common	VCDK	VendingCommonDESKey

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: VendingDefaultDESKey

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: VendingUniqueDESKey

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: VendingCommonDESKey

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 25 and inherit their type from that of the VendingKey, from which they are derived.

Table 25 – 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 10 and Table 26 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.

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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens or via a manufacturer proprietary loading mechanism that utilizes the Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens. The payment meter should only accept the DDTK, DUTK or DCTK encrypted under the DITK supplied by the manufacturer in the Set1stSectionDecoderKey and Set2ndSectionDecoderKey token 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 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;
- 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 Set1stSectionDecoderKey and Set 2ndSectionDecoderKey tokens, or via a manufacturer's proprietary loading mechanism that utilizes the Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens. 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens, or via a manufacturer’s proprietary loading mechanism that utilizes the Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens. 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 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens, or via a manufacturer's proprietary loading mechanism that utilizes the Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens. 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

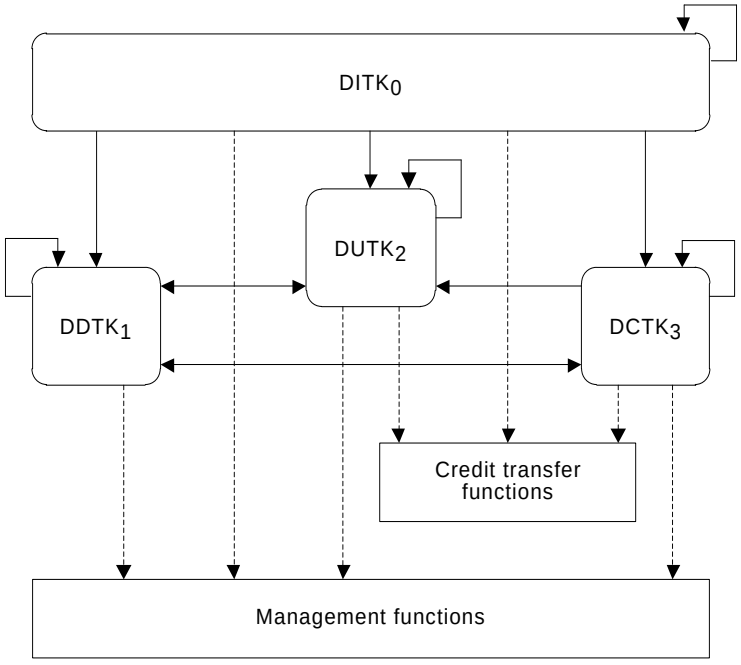


Figure 10 – DecoderKey changes – state diagram

Figure 10 illustrates the KT states that a DecoderKey may assume from time to time.

Where one key is used to encrypt another key (as in the Set1stSectionDecoderKey and Set2ndSectionDecoderKey token pair), 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey token pair. 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 26 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey token 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 26 – 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.

6.5.2.5 KeyRevisionNumber (KRN)

A KRN is associated with each VendingKey and a corresponding SGC by the KMS, and defines the revision or sequence of the VendingKey within the SupplyGroup to which it corresponds. It is a single decimal digit with a range of 1, 2..9. The KRN assigned to the first VendingKey for a SupplyGroup is 1. Successive VendingKeys are allocated successive revision numbers until revision number 9, at which stage the sequence begins at 1 again; in other words, at any given moment, there may be no more than 9 successive VendingKey revisions present for a given SupplyGroup. A KRN is also associated with each DecoderKey, and corresponds to that of the VendingKey from which it is generated.

The KRN is associated with each SupplyGroup by the KMS, and defines the current VendingKey revision and also the current DecoderKey revision, at which all payment meters within the SupplyGroup should be set. For any given payment meter, the SGC and KRN uniquely identify the revision of DecoderKey that it contains. 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 vending KRN for the same SGC as recorded in the management system, this condition shall be corrected by means of an appropriate change of the DecoderKey.

A payment meter is required to store the KRN that corresponds to its current DecoderKey, as passed in the Set1stSectionDecoderKey and Set2ndSectionDecoderKey token pair (see also 7.3.2).

The concept of key revision only applies to vending key types and decoder key types. A DITK shall not be associated with a KRN.

For a given SupplyGroup there shall be a maximum of two active VendingKeys in the POS namely the CurrentKey and the OldKey. The OldKey will only be used to encrypt key change tokens to CurrentKey. The CurrentKey will be used to encrypt all tokens, apart from key change tokens to OldKey.

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 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens respectively (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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey token pair 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey token pair. 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.2.4 for Code of practice on managing this data element.

6.5.3 DecoderKey generation

6.5.3.1 PANBlock construction

The 64-bit PANBlock is constructed from data elements extracted from the MeterPAN in the APDU as defined in Table 27 and Table 28.

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

Table 27 – 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 28 – 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

Where the IIN is 6 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 IIN is 4 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 (for example, for a IIN of 600727, the PANBlock is 0072700000000000).

6.5.3.2 CONTROLBlock construction

The 64-bit CONTROLBlock is constructed from the data elements in the APDU as defined in Table 29 and Table 30.

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

Table 29 – 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 30 – Data elements in the CONTROLBlock

Digit	Name	Format	Reference
C	KT digit	Range 0 to 3 hex per digit, 4 to F hex = reserved	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 standard 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 31.

Table 31 – 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 32.

Table 32 – 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 11.

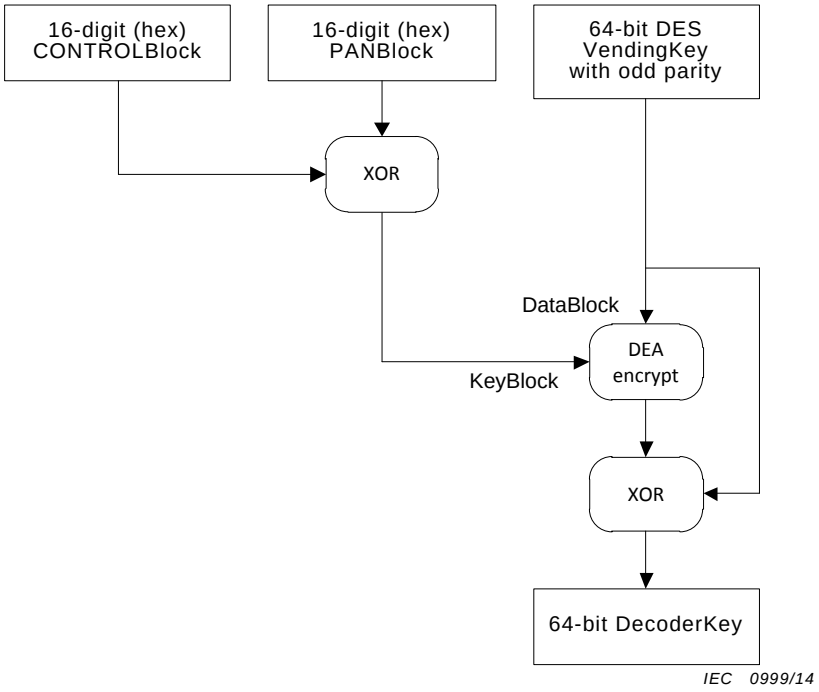


Figure 11 – DecoderKeyGenerationAlgorithm01

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 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 12.

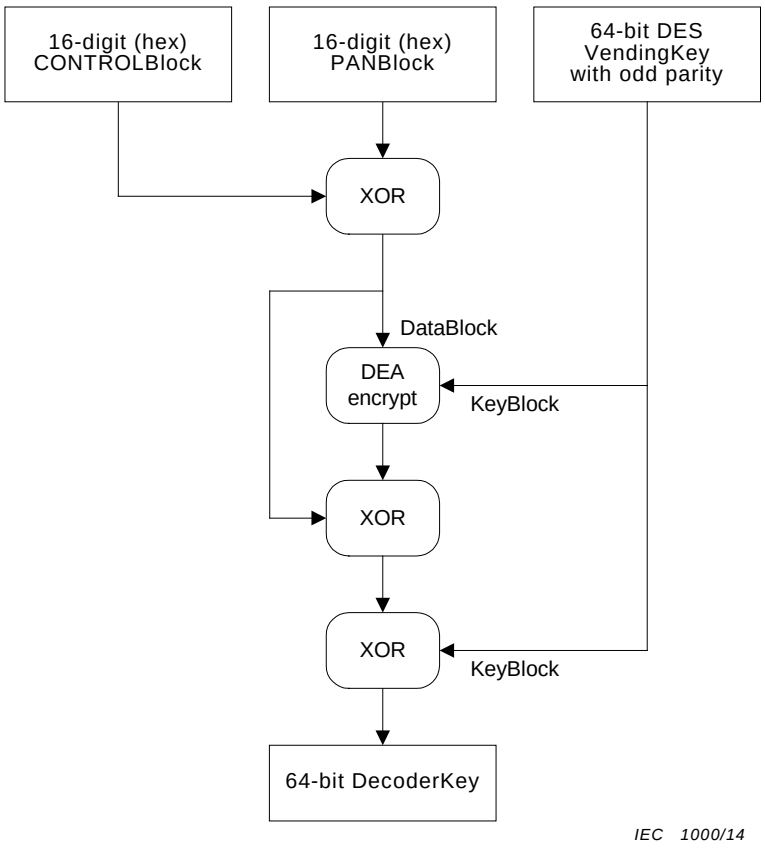


Figure 12 – DecoderKeyGenerationAlgorithm02

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

The DecoderKeyGenerationAlgorithm03 may be used for all payment meters that do not meet the criteria for selecting DecoderKeyGenerationAlgorithm01. The POSApplicationProcess gives the appropriate directive by means of the DKGA code in the APDU.

The DecoderKey is diversified from two 64-bit DES VendingKey values.

The process flow for the DKGA03 is shown in Figure 13.

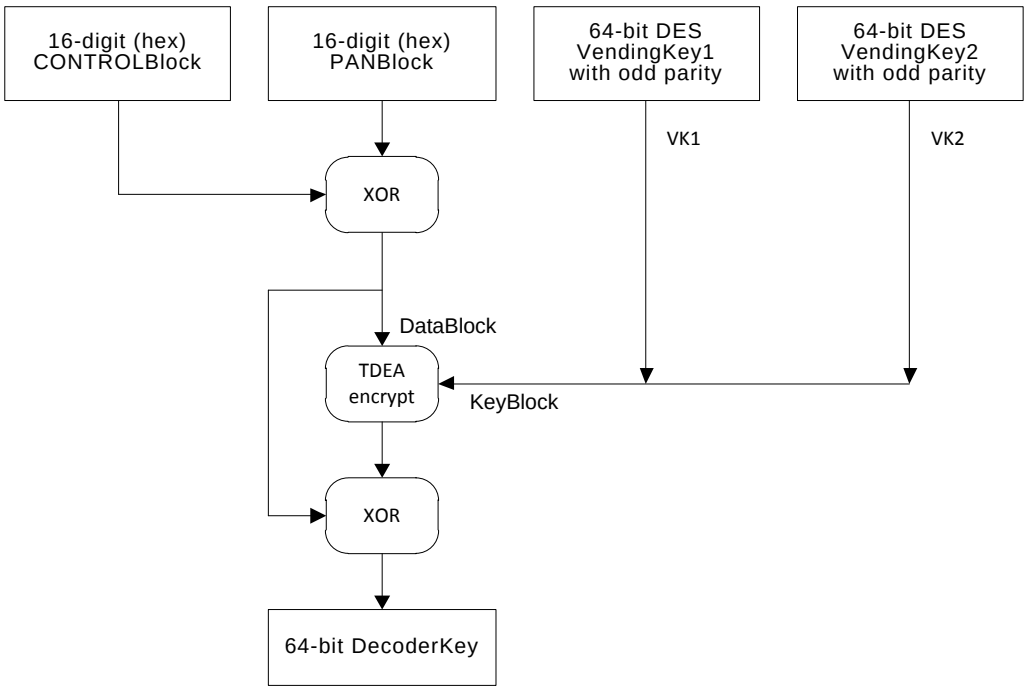


Figure 13 – DecoderKeyGenerationAlgorithm03

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

Encryption is TDEA in accordance with FIPS PUB 46-3, triple DES in ECB mode, using two 64-bit DES VendingKey values VK1 and VK2 with odd parity.

The operation is: encrypt with VK1, decrypt with VK2, encrypt with VK1.

6.5.4 STA: EncryptionAlgorithm07

6.5.4.1 Encryption process

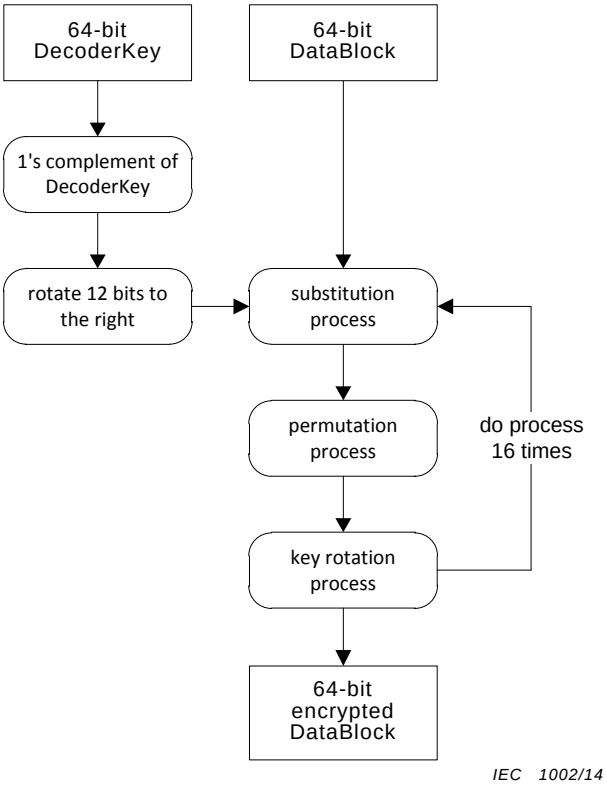


Figure 14 – STA: EncryptionAlgorithm07

The Standard Transfer Algorithm encryption process is shown in Figure 14, 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 15.

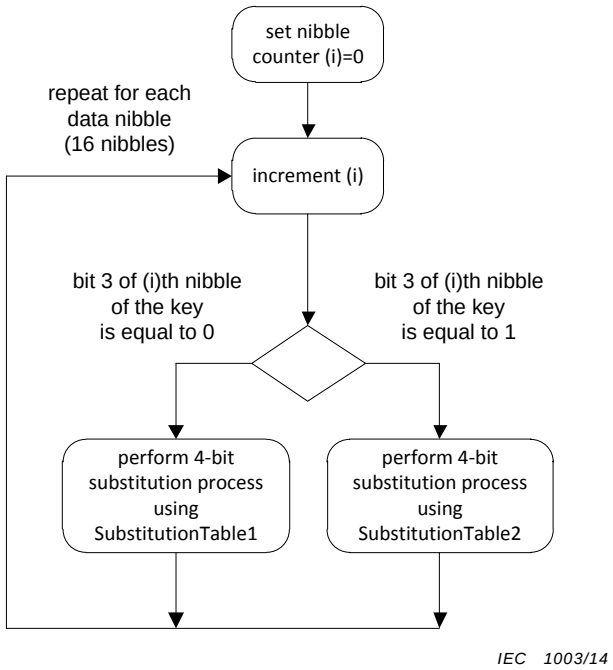


Figure 15 – STA encryption substitution process

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 33.

Table 33 – 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.5 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 16.

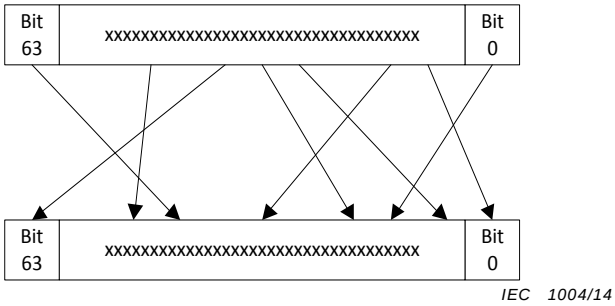


Figure 16 – STA encryption permutation process

A sample permutation table is given in Table 34.

Table 34 – 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.5 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 17.

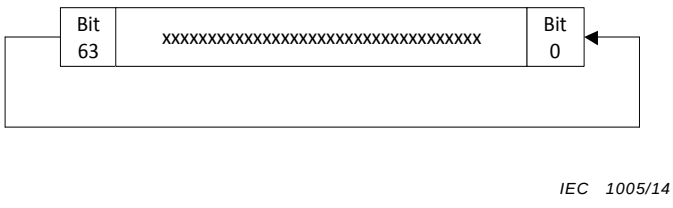


Figure 17 – STA encryption DecoderKey rotation process

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 18.

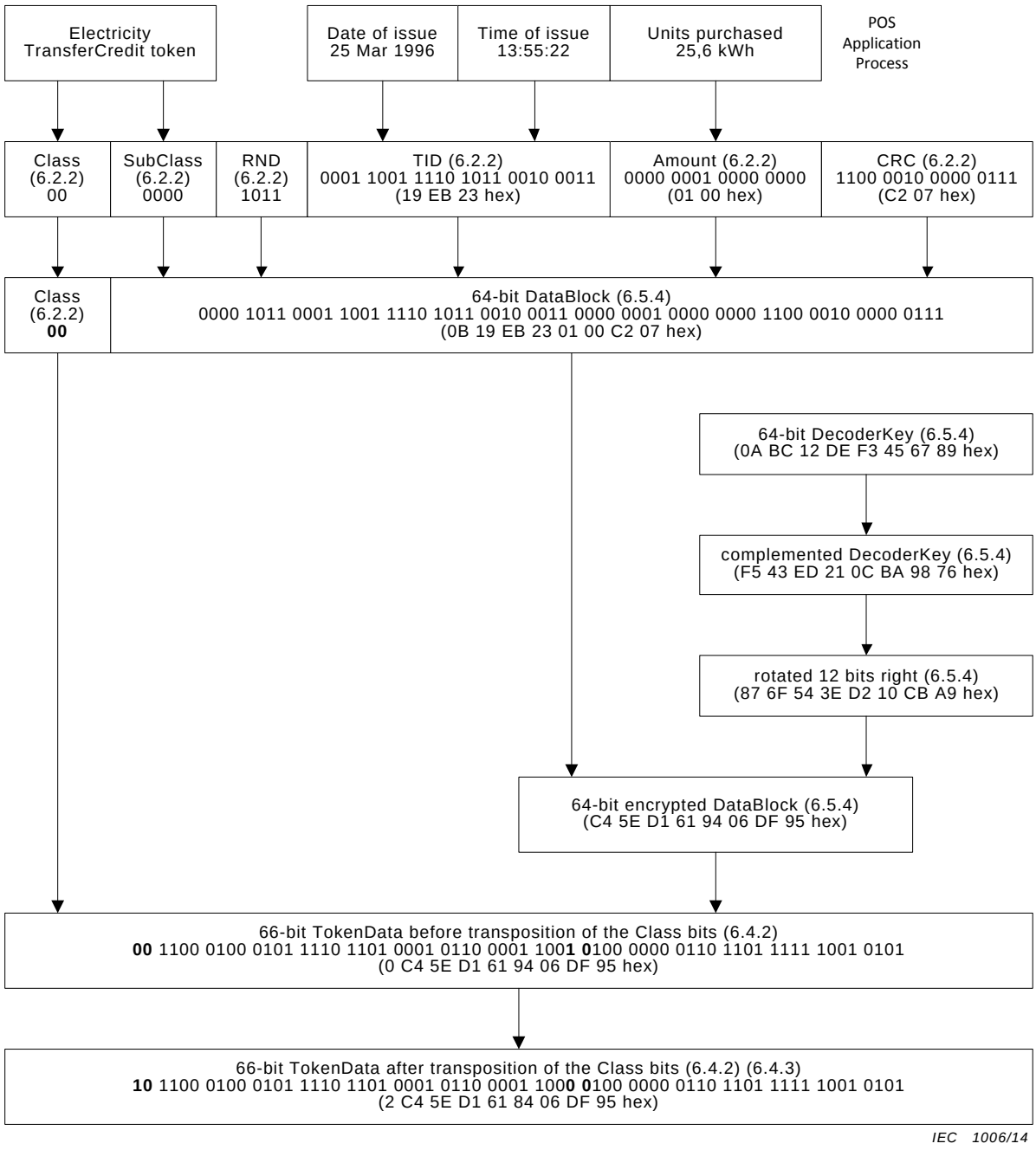


Figure 18 – STA encryption worked example for TransferCredit token

6.5.5 DEA: EncryptionAlgorithm09

The encryption process using the DEA is shown in Figure 19.

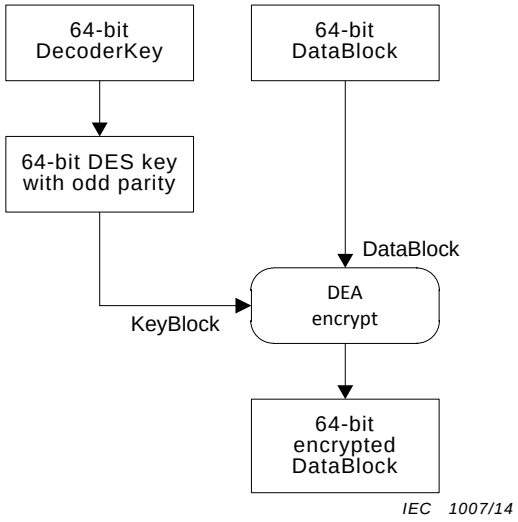


Figure 19 – DEA: EncryptionAlgorithm09

The DEA is a 64-bit block cipher in accordance with FIPS PUB 46-3 operating in ECB mode. The POSApplicationProcess gives the appropriate directive by means of the EA code in the APDU.

The 64-bit DecoderKey is produced with DecoderKeyGenerationAlgorithm02 or with DecoderKeyGenerationAlgorithm03 (see 6.5.3.4 and 6.5.3.5).

The DecoderKey is converted into a 64-bit DES Key with odd parity in accordance with FIPS PUB 46-3 by changing every eighth bit into a parity bit, starting with the least significant bit. Thus, bit 0, bit 8, bit 16, bit 24, bit 32, bit 40, bit 48 and bit 56 are converted into parity bits, where bit 0 is the least significant bit.

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

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 35.

Table 35 – 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 36.

Table 36 – 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 7.3.6) passed or failed, in order that the MeterApplicationProcess can respond appropriately. Possible values are given in Table 37.

Table 37 – 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	7.3.6
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	7.3.6
UsedError	The TID value as recorded in the token is already recorded in the memory store of the payment meter	boolean	7.3.6
KeyExpiredError	The TID value as recorded in the token is larger than the KEN stored in the payment meter memory	boolean	7.3.6
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	7.3.6

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 38.

Table 38 – 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 pair 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 pair 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 implemented	boolean	8.2

7.2 APDUExtraction functions

7.2.1 Extraction process

The process of extracting the APDU from the TCDU is shown in Figure 20.

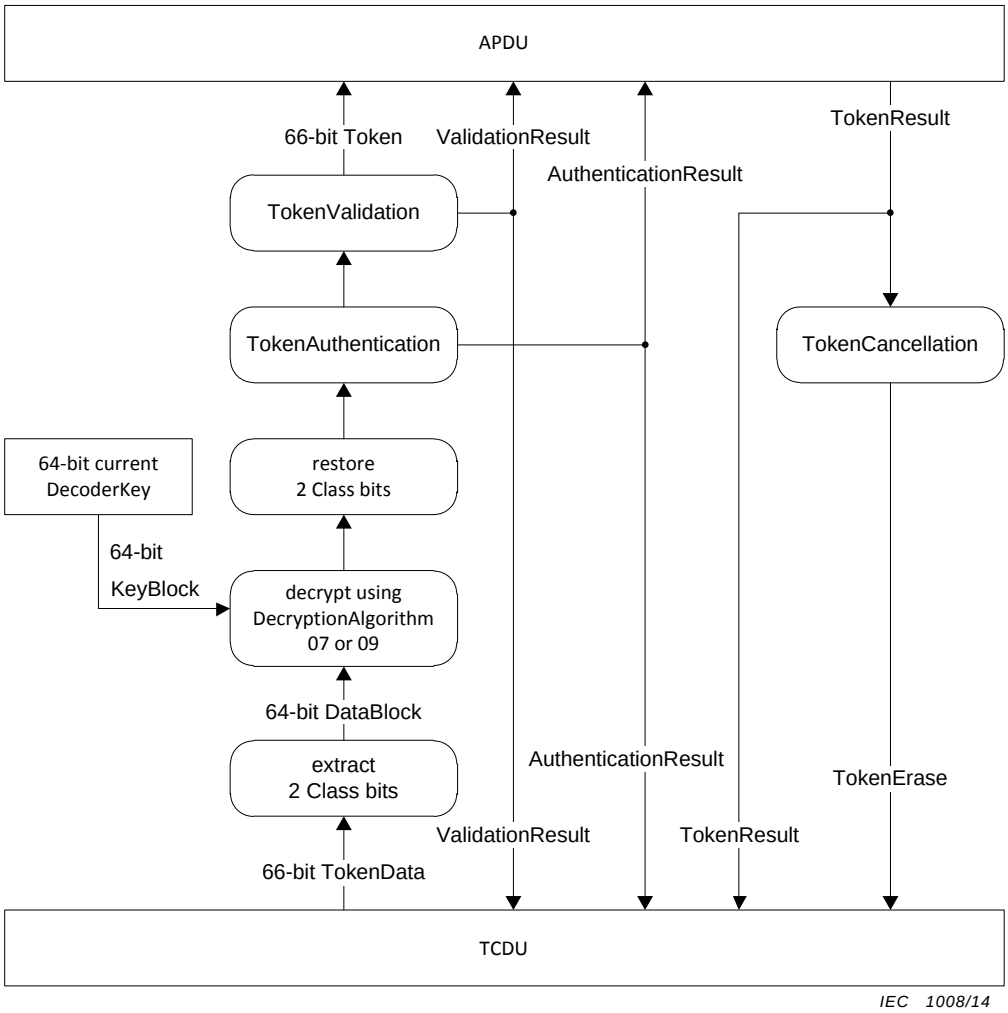


Figure 20 – APDUExtraction function

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 21 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.

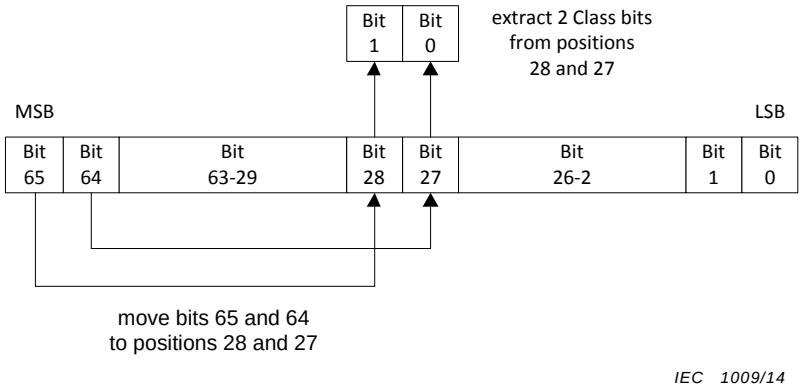


Figure 21 – Extraction of the 2 Class bits

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):

00 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens (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 7.3.6 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 7.3.7 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens

This is the transfer function from the TCDU to the APDU and is applicable to the Set1stSectionDecoderKey and Set2ndSectionDecoderKey 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey 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 or 2ndKCT (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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens 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).

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 39 in secure non-volatile memory.

Table 39 – 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
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.

7.3.3 STA: DecryptionAlgorithm07

7.3.3.1 Decryption process

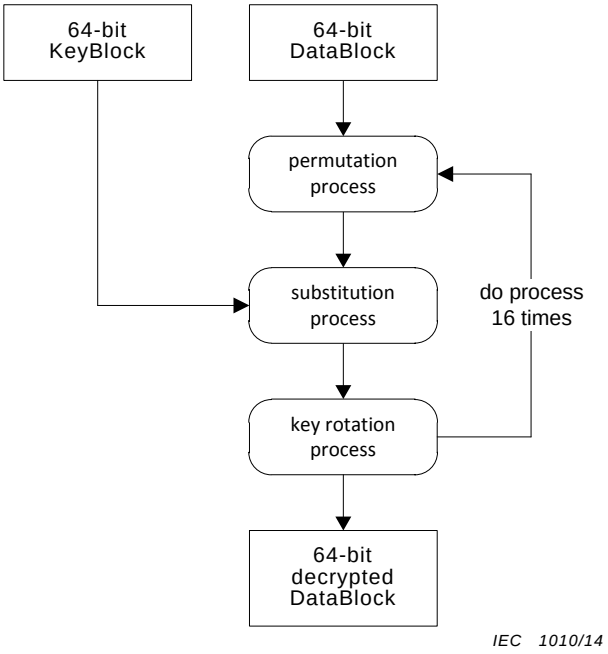


Figure 22 – STA DecryptionAlgorithm07

The Standard Transfer Algorithm decryption process is shown in Figure 22, 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 23.

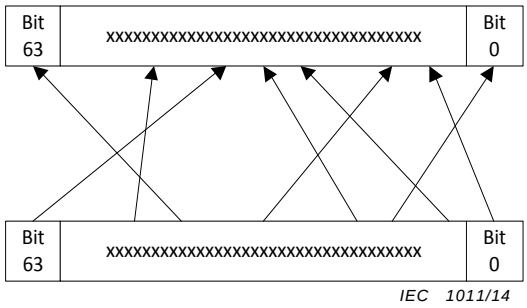


Figure 23 – STA decryption permutation process

A sample permutation table is given in Table 40.

Table 40 – 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.5 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 24.

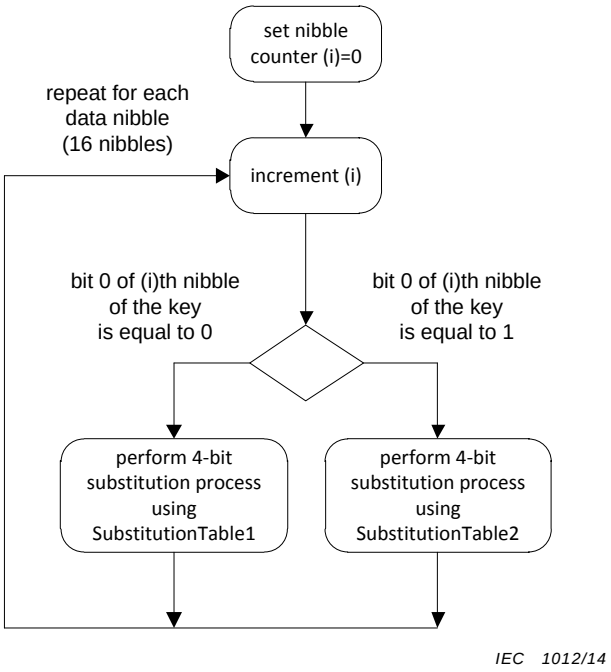


Figure 24 – STA decryption substitution process

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 41.

Table 41 – 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.5 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 25.

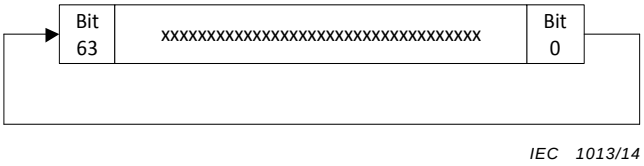


Figure 25 – STA decryption DecoderKey rotation process

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 26.

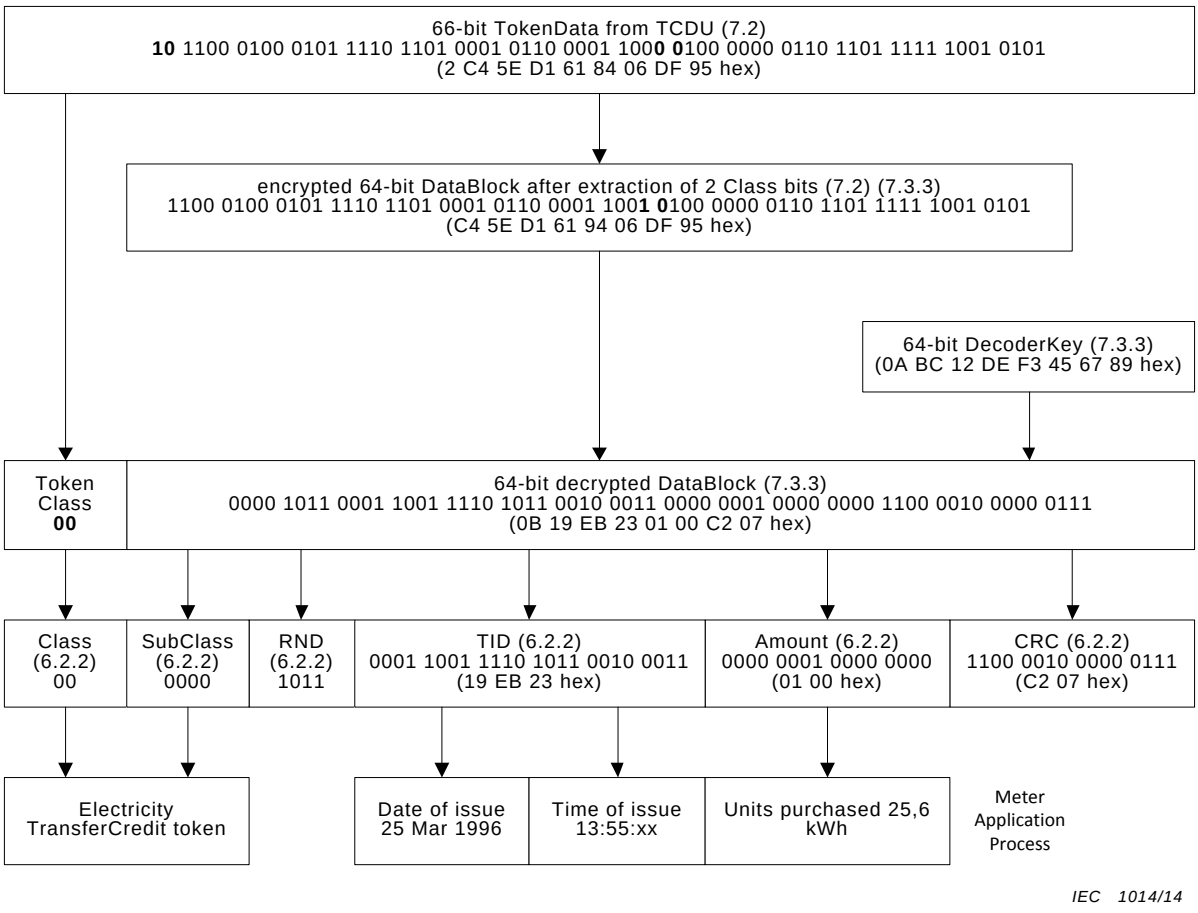


Figure 26 – STA decryption worked example for TransferCredit token

7.3.4 DEA: DecryptionAlgorithm09

The decryption process using the DEA is shown in Figure 27.

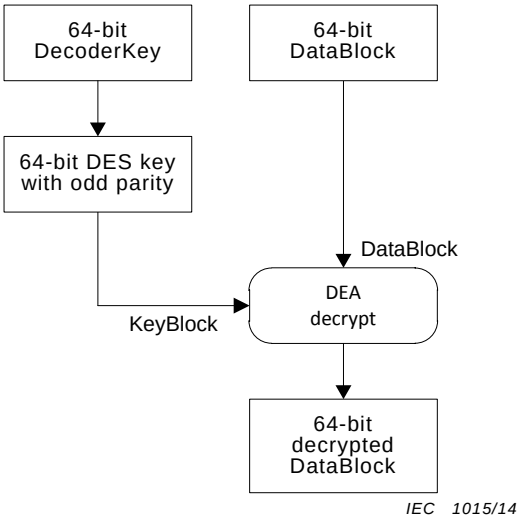


Figure 27 – DEA DecryptionAlgorithm09

The DEA is a 64-bit block cipher in accordance with FIPS PUB 46-3 operating in ECB mode.

The DecoderKey is converted into a 64-bit DES Key with odd parity in accordance with FIPS PUB 46-3 by changing every eighth bit into a parity bit, starting with the least significant bit. Thus, bit 0, bit 8, bit 16, bit 24, bit 32, bit 40, bit 48 and bit 56 are converted into parity bits, where bit 0 is the least significant bit.

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

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

7.3.5 TokenAuthentication

Validating the CRC 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 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.

7.3.6 TokenValidation

Class 0 and Class 2 tokens shall primarily be validated against the TID encoded in the token, except for Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens.

Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens are validated by the MeterApplicationProcess once the payment meter has read both tokens and combined them into the new DecoderKey. See 8.2 for acceptance and rejection requirements of the Set1stSectionDecoderKey and Set2ndSectionDecoderKey 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey 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 7.3.7), 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.

A payment meter loaded with a DDTK value shall accept all the relevant "non-meter-specific management tokens" (Class 1 tokens) as well as Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens encrypted under a DDTK.

7.3.7 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey 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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey token pair, the TID memory store shall remain unchanged, unless the RolloverKeyChange (see 6.3.18) 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.

NOTE A suggested method is for the manufacturer to 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 Set1stSectionDecoderKey token data and Set2ndSectionDecoderKey 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 Set1stSectionDecoderKey or a Set2ndSectionDecoderKey token, whichever is inserted first.

The Set1stSectionDecoderKey or a Set2ndSectionDecoderKey token, 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.

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.

The payment meter shall still accept tokens when in the power limiting or tampered state.

Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens are validated by the MeterApplicationProcess once the payment meter has read both tokens 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 pair shall not indicate Accept, but it shall indicate 1stKCT if it is a Set1stSectionDecoderKey or 2ndKCT if it is a Set2ndSectionDecoderKey token; this indicates provisional acceptance until the second token of the key change token pair is also accepted;
- successful processing of the second entered token of a key change token pair 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;
- 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).

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

In the case where the decoder part is separate from the TokenCarrier interface where the user presents the TokenCarrier to the payment meter, then it shall be possible for the user to determine the DRN and the EA code from the user interface on demand.

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.

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.

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 uitilities 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 standard.

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 overcurrent 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.

The present value in the Tariff Rate Register shall be replaced with the new rate.

8.9 Set1stSectionDecoderKey tokens

See 6.2.7 for more detail on the structure of this token.

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

This action is subject to the successful receipt of both the Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens.

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 the Set1stSectionDecoderKey and Set2ndSection DecoderKey tokens 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 a Set1stSectionDecoderKey and Set2ndSectionDecoderKey token and still perform a successful key change.

It shall be possible to enter the same Set1stSectionDecoderKey and Set2ndSectionDecoderKey token 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

See 6.2.8 for more detail on the structure of this token.

The requirements for the processing of the Set2ndSectionDecoderKey tokens are the same as 8.9 above.

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.

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.

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

NOTE This standard 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 standard 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 standard 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 standard, 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 standard, but may require appropriate amendments to other relevant specifications or COP.

The STS entities and services that require maintenance are given in Table 42.

Table 42 – Entities/services requiring maintenance service

Entity/service	Definition origin	Responsible maintenance body	Reference
Product certification	Clause C.10	STSA/CA	10.2.1
DSN	6.1.2.3.3 C.3.4	Mfr	10.2.2
RO	6.3.18	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.4	utility	10.2.6
MfrCode	6.1.2.3.2 C.3.3	STSA	10.2.7
Substitution tables	6.5.4.2 7.3.3.3 Clause C.5	STSA	10.2.8
Permutation tables	6.5.4.3 7.3.3.2 Clause C.5	STSA	10.2.9
SGC	6.1.6 C.2.2	STSA/KMC	10.2.10
VendingKey	6.5.2.2 Clause 9 C.2.2	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 30	STSA/KMC	10.2.13
KEN	6.1.10 6.5.2.6 C.2.4	STSA/KMC	10.2.14
KEK	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	Mfr/KMC	10.2.19
CMAC	Annex B Table B.2	Mfr/KMC	10.2.20

Entity/service	Definition origin	Responsible maintenance body	Reference
IIN	6.1.2.2 C.3.2	ISO/IEC	10.3.1
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 13 Table 14	STSA/IEC	10.3.5
TokenSubClass	6.3.3 Table 14	STSA/IEC	10.3.6
InitiateMeterTest/Display ControlField	6.3.8 Table 22	STSA/IEC	10.3.7
RegisterToClear	6.3.13 Table 23	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.8	STSA/(IEC)	10.3.13
Companion Specification	5.5 Clause C.8	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 assure access to product certification services to users of the STS.

It shall also assure that such service providers are duly accredited and authorized to provide this service and that they comply with the requirements of this standard 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 base date.

10.2.4 TI maintenance

The utility shall manage the operational use of this element.

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 33 and Table 41 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 34 and Table 40 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 International Standard 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 30.

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 standard.

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 KEK 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 standard 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.

10.2.20 CMAC maintenance

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

The registration transaction of a CMAC value is typically conducted between the CM manufacturer and the KMC service provider, and then it remains in the operations domain of the two parties.

The STS Association, as a registered Registration Authority with the IEC, shall ensure the compliance of such manufacturers and service providers to the requirements and recommendations given in this standard and any other relevant COP.

10.3 Standardisation

10.3.1 IIN maintenance

This standard defines a constant value for electricity payment meters worldwide.

ISO may issue different values for other services upon application by service providers.

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

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 standard 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 standard.

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 standard.

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 13 and Table 14.

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 standard.

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 14.

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 standard.

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 22.

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 standard.

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 23.

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 standard.

10.3.9 STS base date 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 base date value shall require a revision/amendment of this standard.

10.3.10 Rate maintenance

This data element is presently reserved for future definition.

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 standard.

10.3.11 WMFactor maintenance

This data element is presently reserved for future definition.

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 standard.

10.3.12 MFO maintenance

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

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 standard 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 standard 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.

Annex A
(informative)

Guidelines for a KeyManagementSystem (KMS)

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

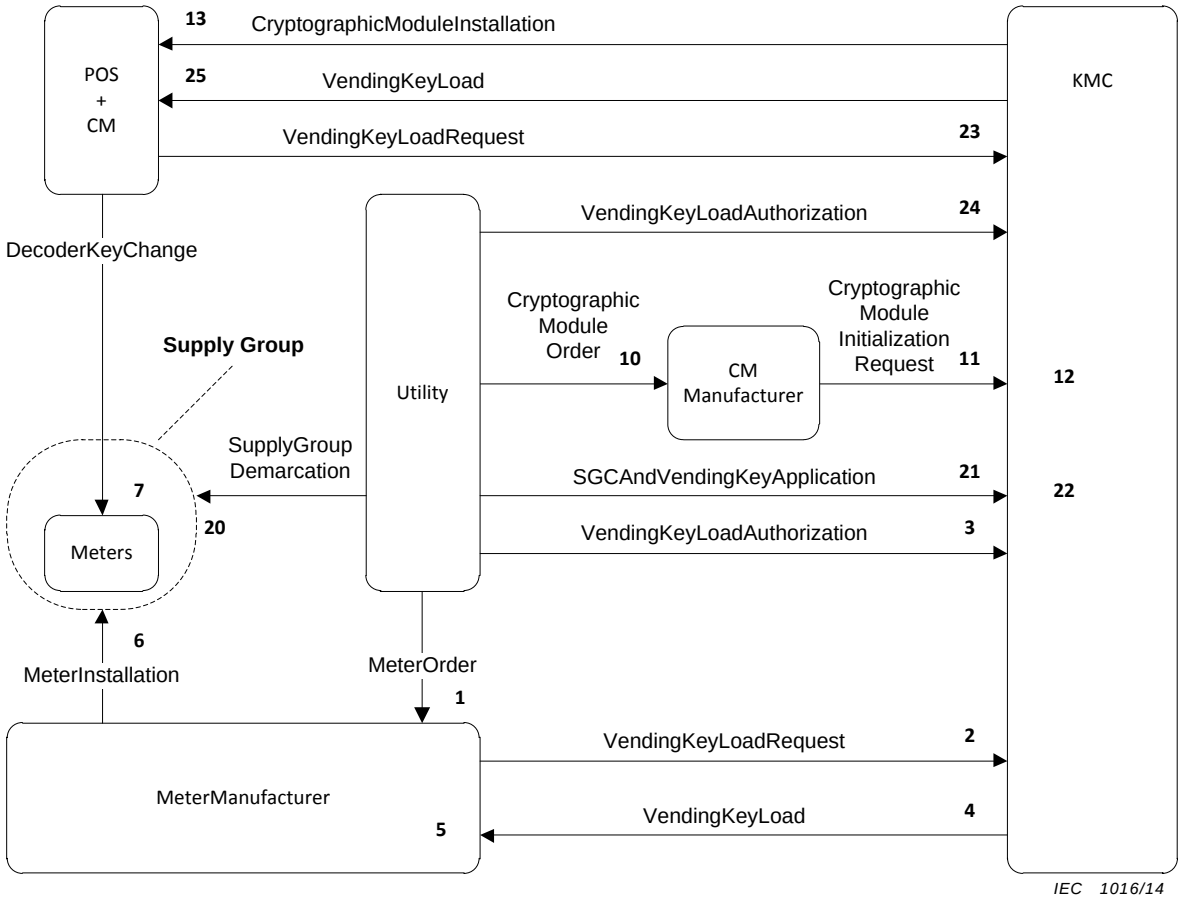


Figure A.1 – KeyManagementSystem and interactive relationships between entities

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

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)
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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens). See also process 23 to 25 below regarding VendingKey loading

The CryptographicModule processes are given in Table A. 3.

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	CryptographicModuleInitialisationRequest The CryptographicModule is sent to the KMC to be initialised with secret key values, which will subsequently be utilized for securely distributing VendingKey values from the KMC to the CryptographicModule
12	CryptographicModuleAuthenticationAndInitialization The KMC checks that the CryptographicModule is authentic and then initialises it with secret key values, which will subsequently be utilized for securely distributing VendingKey values from the KMC to the CryptographicModule (see KEK in Annex B)
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 or common) and associated VendingKey of a specified type (VUDK or VCDK; 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 or VCDK). 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.2 Code of practice for more information regarding the management of VendingKeys.

See also C.2.2.1 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.

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.

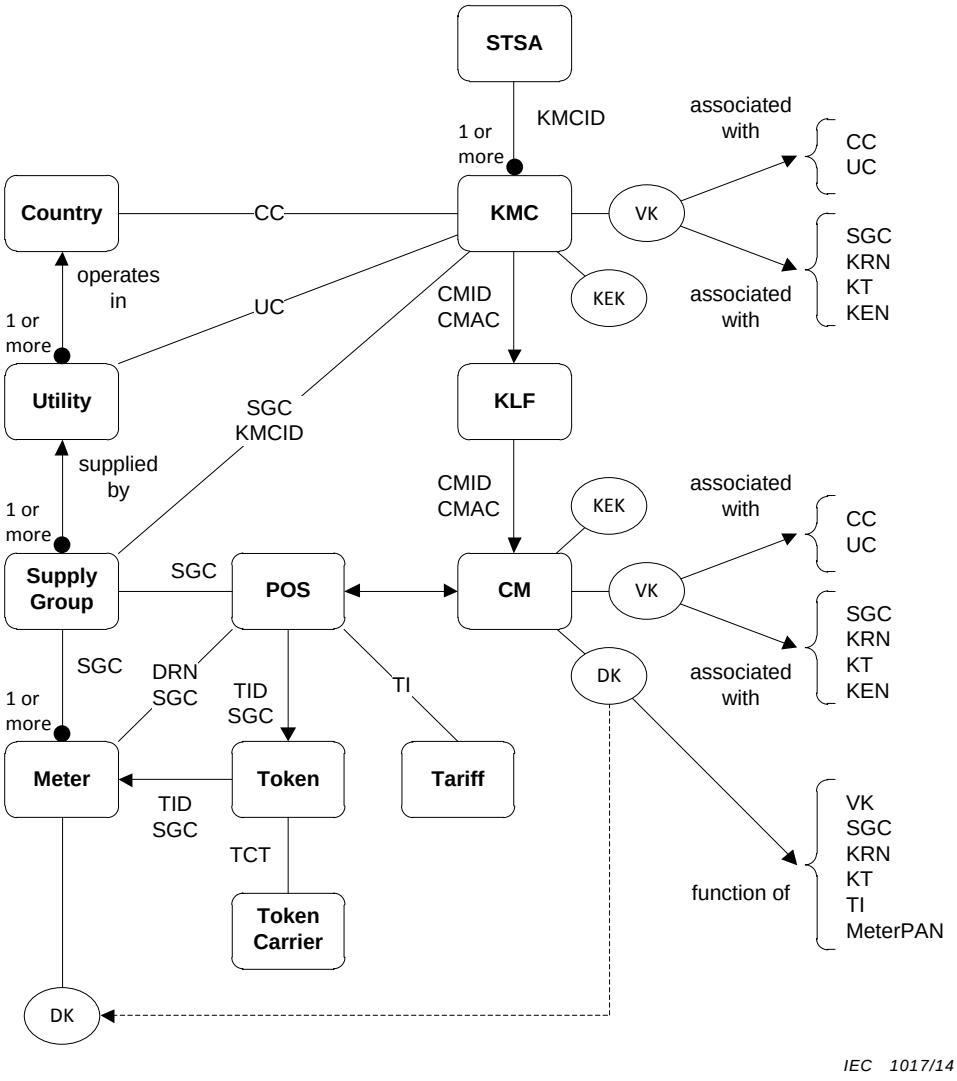


Figure B.1 – Entities and identifiers deployed in an STS-compliant system

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.

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 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 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	9 Annex A Clause C.2
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 POS to generate DecoderKey values from VendingKey values and to generate tokens from DecoderKey values	Annex A
KEK	KeyExchangeKey. A secret dual 64-bit DES key value shared between the KMC and the CM, which is used to encrypt VendingKey values that are distributed by means of the KLF (see KLF above)	x
VK	VendingKey. A 64-bit DES 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
DK	DecoderKey. A 64-bit STA key value or 64-bit DES key value generated as a function of several values: DK = f (VK, SGC, KRN, KT, TI, MeterPAN). 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
CMAC	CryptographicModuleAuthenticationCode A set of secret codes that the KMC and the POS may use to authenticate the CM before entrusting it with other secret values. Typical examples are DeviceAuthenticationCode and FirmwareAuthenticationCode	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
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

Identifier	Context	Reference
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 Set1stSectionDecoderKey and Set2ndSectionDecoderKey tokens	6.1.10

Annex C
(informative)

Code of practice for the implementation of STS-compliant systems

C.1 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 can be contacted at the address given in the introduction to this standard. E-mail is the preferred mechanism for correspondence with the Association.

C.2 Key management

C.2.1 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 standard.

C.2.2 SupplyGroupCode and VendingKey distribution

C.2.2.1 Data elements associated with a SGC

(See also 6.1.6).

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

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

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

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.

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.2.2.2 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 here, 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.2.3 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, KEK, FAC, DAC).	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.2.4 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 standard the key expiry option for DecoderKeys in payment meters had not been implemented in any STS-compliant installation.

C.3 MeterPAN

C.3.1 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.3.2 IssuerIdentificationNumbers

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

C.3.3 ManufacturerCodes

(See also 6.1.2.3.2.)

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 viewed on the STS web site or obtained from the STS Association via any of the contact routes listed above.

C.3.4 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 standard.

C.4 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.5 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 33, Table 34, Table 40 and Table 41) 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.6 EA codes

(See also 6.1.5).

As this standard 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.

C.7 TokenCarrierType codes

(See also 6.1.3).

As this standard 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.

C.8 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 201-15.1.0 (see Bibliography) for an example of a companion specification.

C.9 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 shall change 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 standard, 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.10 STS-compliance certification

C.10.1 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.10.2 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.10.3 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.11 Procurement options for users of STS-compliant systems

This standard 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.

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; • DEA code 09. 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	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; • TDEA (DKGA03);future systems	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: • 001; South African KMC currently in operation; • xxx; future possible KMC of choice or relevance	Annex B
SGC	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	6.1.6

Item	Context	Reference
TI	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); • don't 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	6.1.7
KRN	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	6.1.8
KT	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	6.1.9
KEN	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	6.1.10
DecoderKey expiry	Whether the DecoderKeys should expire or not. Options: • shall not expire (this is the current recommended practice); • shall expire. (this implies periodic DecoderKey changes)	6.1.10
VendingKey expiry	Whether the VendingKeys should expire or not. Options: • shall not expire (this is the current recommended practice); • shall expire (this is not currently supported)	6.1.10
Meter dispatching key	Which DecoderKey type the MeterManufacturer should load into the payment meter. Options: • DDTK (manufacturer Default key); • DUTK (utility Unique key); • DCTK (utility Common key)	6.1.6

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) • Set1stSectionDecoderKey; • Set2ndSectionDecoderKey; • 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; • 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

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 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 shall provide a copy of the particular product's STS certificate of compliance as issued by the relevant CertificationAuthority	C.10

C.12 Management of TID Rollover

C.12.1 Introduction

The Token Identifier 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 1st January 1993. The incrementing of the 24 bit field 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.

In order to overcome this problem all meters will require key change tokens with the roll over bit set. In addition to this, the base date of 01/01/1993 will be required to be changed to a

later date. This process will force the meters to reset the TID stack to 0. To avoid previously played tokens from being accepted by the meter due to the TID stack reset, the key change process shall introduce into the meter, a new decoder key.

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

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.12.2 Overview

C.12.2.1 General

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

The current problem is that the TID is generated using a base date of 01/01/1993. The 24 bit value will reach a roll over point in 24/11/2024. In order to manage this, a new base date will need to be created for which all tokens generated will restart with a TID value of 0. Although this, in effect, shifts the problem, more than one base date in staggered intervals can be used.

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;
- Security modules;
- Vending systems;
- Meter upload files;
- Meter manufacturing equipment;
- Meters.

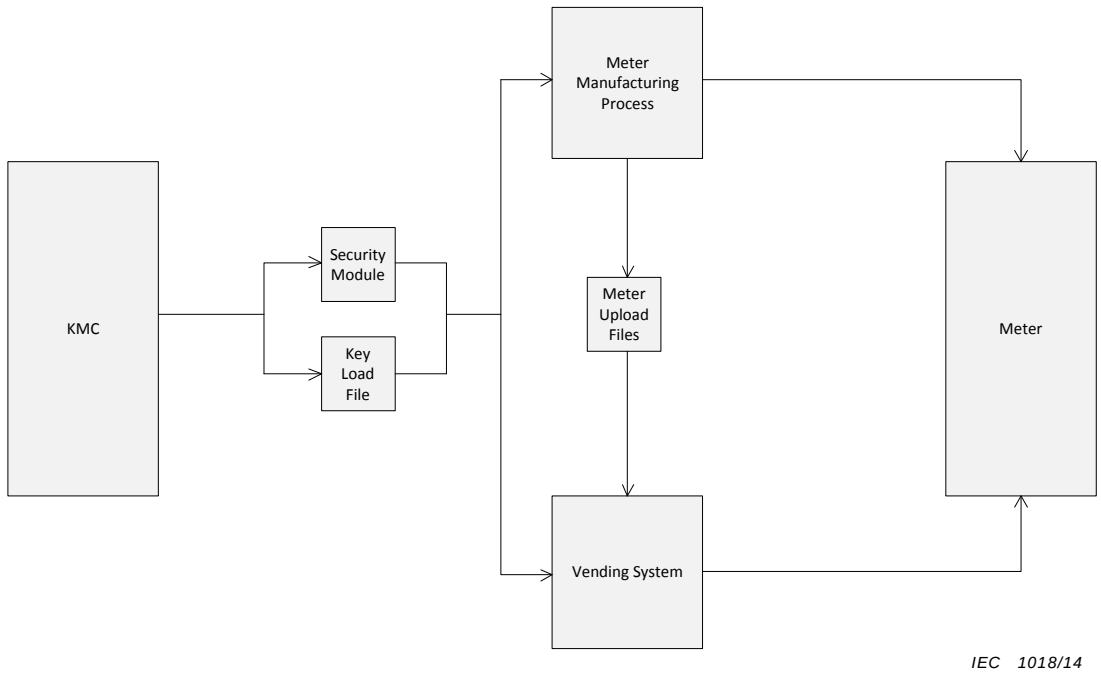


Figure C.1 – System overview

C.12.2.2 Key management centre (KMC)

The KMC is used to generate and load vending keys (Vk) into a security module. The KMC also generates a Key Load File (KLF) which contains the key load data for a specific security module to allow a vending system to load Vk into the security module attached to the system. Currently the Key Revision Number (KRN) for any Vk is 1.

In order to manage the generation of tokens for a specific base date, the vending system requires the KMC to create a new Vk for each base date interval. The new Vk will be created by incrementing the KRN. Associated with each Vk in the KLF will be the selected base date. Two base dates are supported; namely 01/01/2014 and 01/01/2035. Only two are required as 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.12.2.3 Security Module

The security module will be required to generate key change tokens from a Vk on one base date to a Vk on a new base date. The firmware of the SM will be changed to allow implementation of the rollover functionality.

C.12.2.4 Vending System

The vending system will be required to manage, with each Vk loaded into a security module, an associated base date. This base date will be retrieved from the key load file generated at the KMC. In addition the vending system shall associate each meter registered, with the Vk base date from which it was coded.

Once a new Vk is made available, the vending system shall allow for the management of the change process whereby a meter or group of meters can be scheduled for key roll over. In doing so, the affected meters will undergo a key change with 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 base date.

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

C.12.2.5 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 a Vk with the latest base date and therefore each meter record in the meter upload file will be required to include the base date for which it was coded.

C.12.2.6 Manufacturers equipment

All meters leaving the factory shall be coded using a Vk with the current (latest) base date unless otherwise agreed between the utility and the manufacturer. With the two base dates chosen , namely 2014 and 2035, all meters coded before 2014 will be coded using the base date of 1993. All meters coded between 2014 and 2035 shall be coded with the base date of 2014 and all meters coded after 2035 will utilize a base date of 2035, unless otherwise agreed between the utility and the manufacturer.

C.12.2.7 Meters

All STS compliant meters shall support key roll over.

C.12.3 Impact analysis

C.12.3.1 General

The following areas which will be affected by the above process are listed below:

C.12.3.2 Key management centre

- Need to include a base date in the key load file for each Vk;
- Support the selection of predefined base dates when generating Vk;
- Security Modules shall support the key roll over flag.

C.12.3.3 Vending systems

- Associate each Vk for a SGC with a base date as derived from the key load file generated by the KMC;
- Associate each meter with a base date for which the meter is coded. This shall include the extraction of the base date from the Meter Upload File import;
- Allow meters associated with a previous base date to be scheduled individually, in groups or by SGC for a key change to Vk on a new base date and include the key roll over flag.

C.12.3.4 Meter upload files

- Manufacturers to include a base date with each meter record in the file;
- Meter Upload File specifications will need to be updated to reflect the addition of the base date.

C.12.3.5 Manufacturing equipment

- Shall automatically code all meters using the Vk with the latest active base date as agreed with the utility;
- Meters;
- Shall support key roll over.

C.12.4 Base dates

See 6.3.5 above.

C.12.5 Implementation

C.12.5.1 General

Implementation details for manufacturers of meters and vending systems have been outlined in the document body. The sections that follow give basic guidelines for Utilities to follow in the successful implementation of the TID key-change program. Note that Utilities may elect to follow alternative methods of implementation.

C.12.5.2 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 security modules:

a) Secure module firmware has been changed to support the rollover functionality.

b) Vending software suppliers have modified the vending software to recognise the base dates as described in this standard. Once a meter has been key-changed with rollover, this fact shall then be recorded into the vending database.

c) All manufactured meters support the rollover 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 rollover functionality. It is envisaged that the current installed base will no

longer be in service by the time key rollover is required, and that all meters manufactured after the first base date change of 2014, will support the rollover functionality.

C.12.5.3 Process for utilities

A guideline to the process to follow is presented below:

a) Plan the TID rollover program so as to complete the installed base of meters 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.

c) Upgrade all vending installations to software that supports the rollover functionality and relevant database changes.

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 secure modules with rollover functionality through the secure module supplier.

f) Upgrade KMC software, where this is owned by a utility, to cater for multiple base dates.

g) Contact the manufacturer of your meters to confirm whether their meters support key-change with rollover. If not, these meters will have to be replaced in the field with meters that do.

h) Start the key-change process.

C.12.5.4 Key-change process

Various options exist for the physical key-change process:

a) Generate key-change tokens (two 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 user. Explain to the user 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 users and ask them to come in to fetch 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 user and may cause unnecessary concerns.

C.12.5.5 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 stop 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, Secure 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 user, 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, shall be coded 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 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.

Bibliography

ISO 8731-1, *Banking – Approved algorithms for message authentication – Part 1: DEA*

ISO 4909, *Banking cards – Magnetic stripe data content for Track 3*

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 201-15.1.0, *Standard transfer specification (STS) – Companion specification – Meter function object: RegisterTable for electricity payment meters*

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*

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)*

NOTE STS documents are available from the STS Assocation world wide web www.sts.org.za

SOMMAIRE

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

COMPTAGE DE L'ÉLECTRICITÉ – SYSTÈMES DE PAIEMENT –

Partie 41: Spécification de transfert normalisé (STS) – Protocole de couche application pour les systèmes de supports de jeton unidirectionnel

AVANT-PROPOS

1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.

La Norme internationale IEC 62055-41 a été établie par le comité d'études 13 de l'IEC: Mesure de l'énergie électrique, contrôle des tarifs et de la charge.

Cette deuxième édition annule et remplace la première édition, parue en 2007. Cette édition constitue une révision technique. Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

- Le jeton de Classe 2 est étendu de façon à inclure le transfert de crédit pour le gaz et l'eau et les extensions associées dans les jetons display/test.
- MfrCode est étendu de 2 à 4 chiffres.
- Trois dates de référence d'identificateur de jeton sont définies pour des changements de clé plus fréquents avec des procédures de passage par zéro de l'identificateur de jeton (TID).

- Un code de bonnes pratiques pour la gestion des changements de clé par passage par zéro de l'identificateur de jeton (TID) en association avec l'ensemble révisé de dates de référence.
- Des clarifications et des exemples supplémentaires ont été introduits.

Le texte de cette norme est issu des documents suivants:

CDV	Rapport de vote
13/1530/CDV	13/1553/RVC

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/IEC, Partie 2.

Une liste de toutes les parties de la série IEC 62055, publiées sous le titre général *Comptage de l'électricité – Systèmes de paiement*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "http://webstore.iec.ch" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

INTRODUCTION

La série IEC 62055 couvre les systèmes de paiement, englobant les systèmes d'informations des consommateurs, les systèmes de points de vente, les supports de jetons, les compteurs de paiement et les interfaces respectives qui existent entre ces entités. Au moment de la préparation de la présente Norme, l'IEC 62055 comprenait les parties suivantes, sous le titre général, *Équipements de comptage de l'électricité – Systèmes de paiement:*

Partie 21: Cadre pour la normalisation

Partie 31: Exigences particulières - Compteurs statiques à paiement d'énergie active (classes 1 et 2)

Partie 41: Spécification de transfert normalisé (STS) - Protocole de couche application pour les systèmes de supports de jeton unidirectionnel

Partie 51: Spécification de transfert normalisé – Protocole de couche physique pour supports de jeton à carte magnétique et numérique unidirectionnel

Partie 52: Spécification de transfert normalisé – Protocole de couche physique pour support de jeton virtuel bidirectionnel pour raccordement local direct

La série des Parties 4x spécifie les protocoles de couche application et la série des Parties 5x spécifie les protocoles de couche physique.

La Spécification de transfert normalisé (Standard transfer specification - STS) est un protocole de message sécurisé qui permet de transporter des informations entre des équipements de point de vente (Point of sale - POS) et des compteurs de paiement. Elle permet plusieurs types de message tels que les consignes concernant le crédit, la maîtrise de la configuration, l'affichage et les essais. Elle spécifie en outre les dispositifs et les codes de pratique pour permettre la prise en charge de la gestion sécurisée (génération, stockage, retrait et transport) des clés cryptographiques utilisées au sein du système.

Le support de jeton, qui n'est pas spécifié dans la présente Partie de l'IEC 62055, est le dispositif ou support physique utilisé pour transporter les informations, et ce, de l'équipement de POS vers le compteur à paiement. Trois types de supports de jetons sont actuellement spécifiés dans l'IEC 62055-51 et dans l'IEC 62055-52; la carte magnétique, le support de jeton numérique et un support de jeton virtuel, qui ont été approuvés par la STS Association. De nouveaux supports de jeton peuvent être proposés comme nouvelles études par l'intermédiaire des Comités nationaux ou par l'intermédiaire de la STS Association.

Bien que la principale mise en œuvre de la STS se situe dans l'industrie d'alimentation en électricité, elle permet la prise en charge de la gestion d'autres services d'une entreprise de distribution comme l'eau et le gaz. Il convient de noter que certaines fonctionnalités peuvent ne pas s'appliquer dans tous les services d'une entreprise de distribution, un exemple en étant "MaximumPowerLimit" dans le cas d'un compteur d'eau. De même, certaines terminologies peuvent ne pas être appropriées dans des applications hors du domaine de l'électricité, un exemple en étant "Load Switch" dans le cas d'un compteur de gaz. Les révisions futures de la STS peuvent permettre la prise en charge d'autres technologies de supports de jeton comme les cartes intelligentes et les clés à mémoire avec une fonctionnalité bidirectionnelle et permettre une horloge temps réel et des tarifs complexes dans le compteur à paiement.

Toutes les exigences spécifiées dans la présente Norme ne sont pas obligatoires pour une mise en œuvre dans une configuration particulière de système. À titre de lignes directrices, un choix de paramètres de configuration facultatifs est énuméré à l'Article C.11.

La STS Association est enregistrée auprès de l'IEC comme une Autorité d'enregistrement pour fournir des services de maintenance en appui à la STS (voir Article C.1 pour plus d'informations).

La publication de l'IEC 62055-41 Éd. 1 en mai 2007 a conduit à son adoption rapide comme la norme globale préférentielle pour les compteurs de prépaiement dans plusieurs pays membres de l'IEC et dans une majorité de pays membres affiliés à l'IEC. Des compteurs d'électricité pour le prépaiement et leurs systèmes de paiement associés sont maintenant produits, exploités et maintenus dans un écosystème d'entreprises de distribution, de constructeurs de compteurs, d'opérateurs de compteurs, de fournisseurs de système distributeur, d'agents de vente, d'institutions bancaires et d'industries adjacentes. Les intérêts pluripartites sont servis par la STS Association comportant plus de 130 organisations sises dans plus de 24 pays. L'interopérabilité et la conformité au Système de transfert normalisé (STS) sont garanties par des spécifications d'essai de conformité développées et gérées par la STS Association. Une liste complète des services de la STS Association peut être consultée à l'adresse http://www.sts.org.za/.

Initialement développée pour des compteurs d'électricité de prépaiement en Afrique - par l'intermédiaire d'une liaison de type D du groupe de travail WG 15 du Comité d'études 13 de l'IEC avec la STS Association - la présente Norme IEC sert maintenant plus d'utilisateurs en Asie qu'en Afrique, avec un total d'environ 35 millions de compteurs exploités par 400 entreprises de distribution dans 30 pays. La gestion de la technologie a été administrée par la STS Association dans le cadre de l'accomplissement de son rôle d'Autorité d'enregistrement désignée par l'IEC.

Le succès global a engendré un besoin pressant d'étendre la gamme des éléments numériques contenus dans les tableaux de l'IEC 62055-41. En particulier, il est nécessaire d'étendre la plage des numéros de constructeurs au-delà des 99 numéros initialement fournis. En outre, l'application de la norme a été étendue pour permettre des systèmes d'énergies multiples comprenant des compteurs de gaz et d'eau. En conséquence, le besoin existe d'assurer que le contenu de l'IEC 62055-41 est maintenu pour permettre cette croissance du marché et ces extensions à énergies multiples.

Plusieurs corrections et clarifications sont également requises pour réactualiser l'Édition 1 par rapport à la pratique courante. Cela avait été envisagé par le groupe de travail WG 15 du CE 13 lors de sa réunion du 20 septembre 2012 à Londres. Selon l'accord conclu, il convient de réviser l'IEC 62055-41.

Seules les révisions requises les plus urgentes ont été incorporées dans l'Édition 2 en raison des contraintes de temps, mais il est prévu que l'Édition 3 envisage des révisions supplémentaires afin d'incorporer les fonctionnalités suivantes:

- Transfert de devises
- Sécurité renforcée allant de pair avec les pratiques industrielles contemporaines
- Fonctions complexes pleinement harmonisées avec la suite DLMS/COSEM (Device Language Message Specification/Companion Specification for Energy Metering, à savoir "Spécification des messages de langage de dispositif/Spécification d'accompagnement pour le comptage d'énergie
- Système de gestion décentralisée de clés avec une architecture distribuée
- Suite d'essais de certification de conformité conjointement avec la Méthode OC de l'IECEE

La Commission Electrotechnique Internationale (IEC) attire l'attention sur le fait qu'il est déclaré que la conformité avec les dispositions du présent document peut impliquer l'utilisation d'un brevet intéressant l'identificateur de jeton réservé spécial indiqué en 6.3.5.2.

L'IEC ne prend pas position quant à la preuve, à la validité et à la portée de ces droits de propriété.

Le détenteur de ces droits de propriété a donné l'assurance à l'IEC qu'il consent à négocier des licences avec des demandeurs du monde entier, soit sans frais, soit à des termes et

conditions raisonnables et non discriminatoires. À ce propos, la déclaration du détenteur des droits de propriété est enregistrée à l'IEC. Des informations peuvent être demandées à:

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

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

L'attention est d'autre part attirée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété autres que ceux qui ont été mentionnés ci-dessus. L'IEC ne saurait être tenue pour responsable de l'identification de ces droits de propriété en tout ou partie.

L'ISO (www.iso.org/patents) et l'IEC (http://patents.iec.ch) maintiennent des bases de données, consultables en ligne, des droits de propriété pertinents à leurs normes. Les utilisateurs sont encouragés à consulter ces bases de données pour obtenir l'information la plus récente concernant les droits de propriété.

La Commission Électrotechnique Internationale (IEC) attire l'attention sur le fait qu'il est déclaré que la conformité aux dispositions de la présente Norme Internationale peut impliquer l'utilisation d'un service de maintenance concernant la gestion de clé de chiffrement et la pile de protocoles sur lesquels est basée la présente Norme IEC 62055-41 [Voir Article C.1]. L'IEC ne prend pas position quant à la preuve, la validité et la portée de ces services de maintenance.

Le fournisseur du service de maintenance a donné l'assurance à l'IEC qu'il consent à fournir ces services aux demandeurs du monde entier, à des termes et conditions raisonnables et non discriminatoires. À ce propos, la déclaration du fournisseur du service de maintenance est enregistrée à l'IEC. Des informations peuvent être demandées à:

Adresse: The STS Association, P.O. Box 868, Ferndale 2160, Republic of South Africa
Tél.: +27 11 061 5000
Fax: +27 86 679 4500
Email: sts@vdw.co.za
Site web: http://www.sts.org.za

COMPTAGE DE L'ÉLECTRICITÉ – SYSTÈMES DE PAIEMENT –

Partie 41: Spécification de transfert normalisé (STS) – Protocole de couche application pour les systèmes de supports de jeton unidirectionnel

1 Domaine d'application

La présente Partie de l'IEC 62055 spécifie le protocole de couche application de la STS pour transférer des unités de crédit et autres informations de gestion, et ce, d'un système de point de vente (POS) vers un compteur à paiement conforme à la STS dans un système de support de jeton unidirectionnel. À l'origine, elle est destinée à être appliquée avec les compteurs à paiement d'électricité simple tarif utilisant des jetons basés sur l'énergie. Mais elle peut également être appliquée aux systèmes de jeton basés sur la monnaie et pour les services autres que l'électricité.

Elle spécifie:

- une interface POS/support de jeton structurée avec un protocole de couche application et un protocole de couche physique utilisant le modèle OSI comme référence;
- des jetons pour le protocole de couche application pour transférer les divers messages du POS vers le compteur à paiement;
- des fonctions et des processus de sécurité dans le protocole de couche application tels que l'Algorithme de transfert normalisé (Standard Transfer Algorithm) et l'Algorithme de chiffrement de données (Data Encryption Algorithm), y compris la génération et la distribution des clés cryptographiques associées;
- des fonctions et des processus de sécurité dans le protocole de couche application au niveau du compteur à paiement tels que les algorithmes de déchiffrement, l'authentification, la validation et l'annulation de jeton;
- des exigences spécifiques relatives au processus d'application de compteur en réponse aux jetons reçus;
- une méthode pour traiter de la fonctionnalité de compteur à paiement dans le processus d'application de compteur et les spécifications d'accompagnement associées;
- des exigences génériques relatives à un système de gestion de clé conforme à la STS;
- des lignes directrices pour un système de gestion de clé;
- des entités et des identificateurs utilisés dans un système STS;
- le code de bonnes pratiques pour la gestion des changements de clé par passage par zéro de l'identificateur de jeton (TID) en association avec l'ensemble révisé de dates de référence;
- le code de bonnes pratiques et les services de support à la maintenance provenant de la STS Association.

Elle est destinée à être utilisée par les constructeurs de compteurs à paiement tenus d'accepter les jetons conformes à la STS et aussi par les constructeurs de systèmes POS tenus de produire des jetons conformes à la STS. Elle est à lire conjointement avec la série IEC 62055-5x.

Les produits conformes à la STS sont tenus de se conformer à des parties sélectives de la présente Norme internationale seulement, celles-ci sont l'objet du contrat d'achat (voir aussi Article C.11).

NOTE Bien qu'elle ait été mise au point pour les systèmes de paiement pour l'électricité, la norme prend également des dispositions pour les jetons utilisés dans d'autres services d'entreprise de distribution, tels que l'eau et le gaz.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 60050 (toutes les parties), *Vocabulaire Electrotechnique International* (disponible à <http://www.electropedia.org>)

IEC 62051:1999, *Electricity metering – Glossary of terms* (disponible en anglais seulement)

IEC 62055-21:2005, *Electricity metering – Payment systems – Part 21: Framework for standardization* (disponible en anglais seulement)

IEC 62055-31:2005, *Équipements de comptage de l'électricité – Systèmes à paiement – Partie 31: Exigences particulières – Compteurs statiques à paiement d'énergie active (classes 1 et 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* (disponible en anglais seulement)

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* (disponible en anglais seulement)

ISO/IEC 7812-1:2006, *Identification cards -- Identification of issuers -- Part 1: Numbering system* (disponible en anglais seulement)

ISO/IEC 7812-2:2007, *Identification cards -- Identification of issuers -- Part 2: Application and registration procedures* (disponible en anglais seulement)

ANSI X3.92-1981, *American National Standard Data Encryption Algorithm, American National Standards Institute – Data Encryption Algorithm* (disponible en anglais seulement)

FIPS PUB 46-3:1999, *Federal Information Processing Standards Publication – Data Encryption Standard* (disponible en anglais seulement)

3 Termes, définitions et abréviations

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans l'IEC 60050-300, l'IEC 62051, l'IEC 62055-31 ainsi que les suivants s'appliquent.

NOTE Lorsqu'il existe une différence entre les définitions de la présente Norme et celles contenues dans d'autres normes IEC de référence, les définitions de la présente Norme prévalent.

Le terme "compteur" est utilisé de façon interchangeable avec "compteur à paiement", "compteur à prépaiement" et "décodeur", lorsque le décodeur est une sous-partie d'un compteur à paiement d'électricité ou d'un compteur à paiement en plusieurs parties.

Le terme “POS” est utilisé comme synonyme de “CIS” (système d'information des consommateurs), de “SIG” (Système d'informations de gestion) et de “TSP” (Terminal de saisie portable) en ce sens que les jetons peuvent également être générés par ces entités et transférés entre elles et le compteur à paiement.

Le terme “entreprise de distribution” est utilisé pour signifier le fournisseur du service dans un sens général. Dans les marchés libéralisés, la partie contractante réelle qui agit comme le "fournisseur" du service au consommateur peut ne pas être l'entreprise de distribution traditionnelle en tant que telle, mais peut être une tierce partie de fournisseur de service.

3.1.1
spécification d'accompagnement
spécification gérée par la STS Association, qui définit une instance spécifique d'un MeterFunctionObject (voir 5.5 et Article C.8)

3.1.2
décodeur
partie intégrante de la TokenCarrierToMeterInterface d'un compteur à paiement qui accomplit les fonctions du protocole de couche application et qui permet que des transactions à base de jetons aient lieu entre un POS et le compteur à paiement

3.1.3
numéro de série de compteur
nombre associé à la partie métrologique du compteur à paiement

Note 1 à l'article: Dans un compteur à paiement en une seule partie, le DRN (numéro de référence de décodeur) et le numéro de série de compteur peuvent être synonymes, alors qu'ils peuvent être différents dans un compteur à paiement en plusieurs parties.

3.1.4
jeton
sous-ensemble d'éléments de données, contenant une instruction et de l'information qui sont présentes dans l'APDU de la couche application de la POSToTokenCarrierInterface, et qui est également transféré vers le compteur à paiement au moyen d'un support de jeton (l'inverse est également vrai dans le cas d'un jeton envoyé du compteur à paiement vers le POS)

3.1.5
support de jeton
support qui est utilisé dans la Couche physique de la POSToTokenCarrierInterface, sur lequel un jeton est modulé ou codé et qui sert à transporter un jeton du point où il est généré vers le compteur à paiement distant, où il est reçu

3.1.6
système de support de jeton unidirectionnel

système de comptage pour paiement qui utilise des supports de jetons qui transfèrent l'information dans un seul sens – du POS vers le compteur à paiement

3.1.7
transaction à base de jeton
traitement d'un jeton quelconque par le compteur à paiement qui a l'effet matériel sur la quantité, la valeur ou la qualité du service à fournir au consommateur sous le contrôle du compteur à paiement (pour les bonnes pratiques courantes, cela signifie des jetons de Classe 0 et de Classe 2)

3.1.8
pris(e) en charge
aptitude à accomplir une fonction définie

Note 1 à l'article: Si une fonction prise en charge est désactivée, elle reste prise en charge.

3.2 Abréviations

ANSI	American National Standards Institute (Institut national de normalisation des États-Unis d'Amérique)
APDU	ApplicationProtocolDataUnit (Unité de données de protocole d'application)
CA	CertificationAuthority (Autorité de certification)
CC	CountryCode (Code de pays)
CIS	Customer Information System (Système d'information des consommateurs)
CM	CryptographicModule (Module cryptographique)
CMAC	CryptographicModuleAuthenticationCode (Code d'authentification de module cryptographique)
CMID	CryptographicModuleIdentifier (Identificateur de module cryptographique)
COP	Code of practice (Code de bonnes pratiques)
CRC	CyclicRedundancyCode (Code de redondance cyclique)
DAC	DeviceAuthenticationCode (Code d'authentification de dispositif)
DCTK	DecoderCommonTransferKey (Clé de transfert commune de décodeur)
DD	Discretionary Data (Données discrétionnaires)
DDTK	DecoderDefaultTransferKey (Clé de transfert par défaut de décodeur)
DEA	Data Encryption Algorithm (Algorithme de chiffrement de données)
DES	Data Encryption Standard (Norme de chiffrement de données)
DITK	DecoderInitializationTransferKey (Clé de transfert d'initialisation de décodeur)
DK	DecoderKey (Clé de décodeur)
DKGA	DecoderKeyGenerationAlgorithm (Algorithme de génération de clé de décodeur)
DKR	DecoderKeyRegister (Registre de clés de décodeur)
DOE	DateOfExpiry (Date d'expiration)
DRN	DecoderReferenceNumber [(Numéro de référence de décodeur) appelé "numéro de compteur" dans les systèmes utilisés avant la mise au point de la présente Norme]
DSN	DecoderSerialNumber (Numéro de série de décodeur)
DUTK	DecoderUniqueTransferKey (Clé de transfert unique de décodeur)
EA	EncryptionAlgorithm (Algorithme de chiffrement)
ECB	Electronic Code Book (Livre de code électronique)
ETX	ASCII End of Text character (Caractère fin de texte ASCII)
FAC	FirmwareAuthenticationCode (Code d'authentification de firmware)
FIPS	Federal Information Processing Standards (Normes fédérales pour le traitement de l'information)
FOIN	FunctionObjectIdentificationNumber (Numéro d'identification d'objet fonction)
FS	FieldSeparator (Séparateur de champ)
GPRS	General Packet Radio Service (Service général de radiocommunication en mode paquet)
GSM	Global System For Mobile Communications (Système global de communications mobiles)
TSP	HandHeldUnit (Terminal de saisie portable)
IAIN	IndividualAccountIdentificationNumber (Numéro d'identification de compte individuel)
ID	Identification; Identificateur

IIN	IssuerIdentificationNumber (Numéro d'identification d'émetteur)
RNIS	Réseau numérique à intégration de services
ISO	International Standards Organisation (Organisation internationale de normalisation)
ISO BIN	Remplacé par IIN
KCT	KeyChangeToken (Jeton de changement de clé)
KEK	KeyExchangeKey (Clé d'échange de clé)
KEN	KeyExpiryNumber (Numéro d'expiration de clé)
KLF	KeyLoadFile (Fichier de chargement de clés)
KMC	KeyManagementCentre (Centre de gestion de clé)
KMI	KeyManagementInfrastructure (Infrastructure de gestion de clé)
KMS	KeyManagementSystem (Système de gestion de clé)
KRN	KeyRevisionNumber (Numéro de révision de clé)
KT	KeyType (Type de clé)
LAN	Local Area Network (Réseau local)
LRC	LongitudinalRedundancyCheck (Contrôle de redondance longitudinale)
MFO	MeterFunctionObject (Objet fonction de compteur)
Mfr	Manufacturer (Constructeur)
MII	MajorIndustryIdentifier (Identificateur de la principale activité économique)
SIG (MIS)	Système d'informations de gestion (Management Information System)
MPL	MaximumPowerLimit (Limite de puissance maximale)
MPPUL	MaximumPhasePowerUnbalanceLimit (Limite maximale de déséquilibre de puissance de phases)
NIST	National Institute of Standards and Technology (Institut national américain des normes et des technologies)
NKHO	bits NewKeyHighOrder (bits de poids fort de nouvelle clé)
NKLO	bits NewKeyLowOrder (bits de poids faible de nouvelle clé)
NWIP	New Work Item Proposal (Proposition d'étude nouvelle)
OSI	Open Systems Interconnection (Interconnexion de systèmes ouverts)
PAN	PrimaryAccountNumber (Numéro de compte primaire)
PLC	Power Line Carrier (Courants Porteurs en Ligne))
POS	PointOfSale (Point de vente)
PRN	Printer (Imprimante)
RTPC	Réseau téléphonique public commuté
RND	RandomNumber (Nombre aléatoire)
RO	Rollover (Passage par zéro)
SG	SupplyGroup (Groupe d'alimentation/approvisionnement)
SGC	SupplyGroupCode (Code de groupe d'alimentation/approvisionnement)
STA	Standard Transfer Algorithm (Algorithme de transfert normalisé)
STS	Standard Transfer Specification (Spécification de transfert normalisé)
STSA	Standard Transfer Specification Association (STS Association, Association de Spécification de transfert normalisé)
STX	ASCII Start of Text character (Caractère début de texte ASCII)
TCDU	TokenCarrierDataUnit (Unité de données de support de jeton)

TCT	TokenCarrierType (Type de support de jeton)
TDEA	Triple Data Encryption Algorithm (Algorithme de triple chiffrement de données)
TI	TariffIndex (Index de tarifs)
TID	TokenIdentifier (Identificateur de jeton)
UC	UtilityCode (Code d'entreprise de distribution)
VCDK	VendingCommonDESKey (Clé DES commune de vente)
VDDK	VendingDefaultDESKey (Clé DES par défaut de vente)
VK	VendingKey (Clé de vente)
VUDK	VendingUniqueDESKey (Clé DES unique de vente)
WAN	Wide Area Network (Réseau étendu)
XOR	OU exclusif (logique)

3.3 Notation et terminologie

Tout au long de la présente Norme, les règles suivantes sont observées en ce qui concerne la dénomination des termes:

- les noms d'entité, les noms d'élément de données, les noms de fonction et les noms de processus sont traités comme des classes d'objets génériques et reçoivent des noms sous la forme d'expressions dans lesquelles les mots sont en majuscules et aboutés sans espaces. Par exemple: SupplyGroupCode comme nom d'élément de données, EncryptionAlgorithm07 comme nom de fonction et TransferCredit comme nom de processus (voir la note);
- une référence directe (spécifique) à une classe nommée d'objets utilise la forme en majuscules, alors qu'une référence générale (non spécifique) utilise le texte conventionnel, à savoir la forme en minuscules sans espaces. Un exemple de référence directe est "Le SupplyGroupCode (CodeDeGroupeDAlimentation) est lié à un groupe de compteurs", alors qu'un exemple de référence générale est "Un "supply group code" (code de groupe d'alimentation) relie à une clé de vente";
- d'autres termes utilisent les formes abrégées généralement acceptées comme RTPC pour "Réseau téléphonique public commuté".

NOTE La notation utilisée pour la dénomination d'objets a été alignée sur ladite "notation-chameau" utilisée dans les normes du Modèle d'information Commun (Common Information Model - CIM)) établies par le CE 57 de l'IEC, afin de faciliter l'harmonisation et l'intégration futures des normes de système de paiement avec les normes CIM.

4 Conventions de numérotation

Dans la présente Norme, la représentation des nombres en chaînes binaires utilise la convention selon laquelle le bit de poids faible est à droite et le bit de poids fort à gauche.

La numérotation des positions binaires commence par la position binaire 0, qui correspond au bit de poids faible d'un nombre binaire.

Les nombres sont généralement au format décimal, sauf indication contraire. Tout chiffre sans indicateur sous-entend le format décimal.

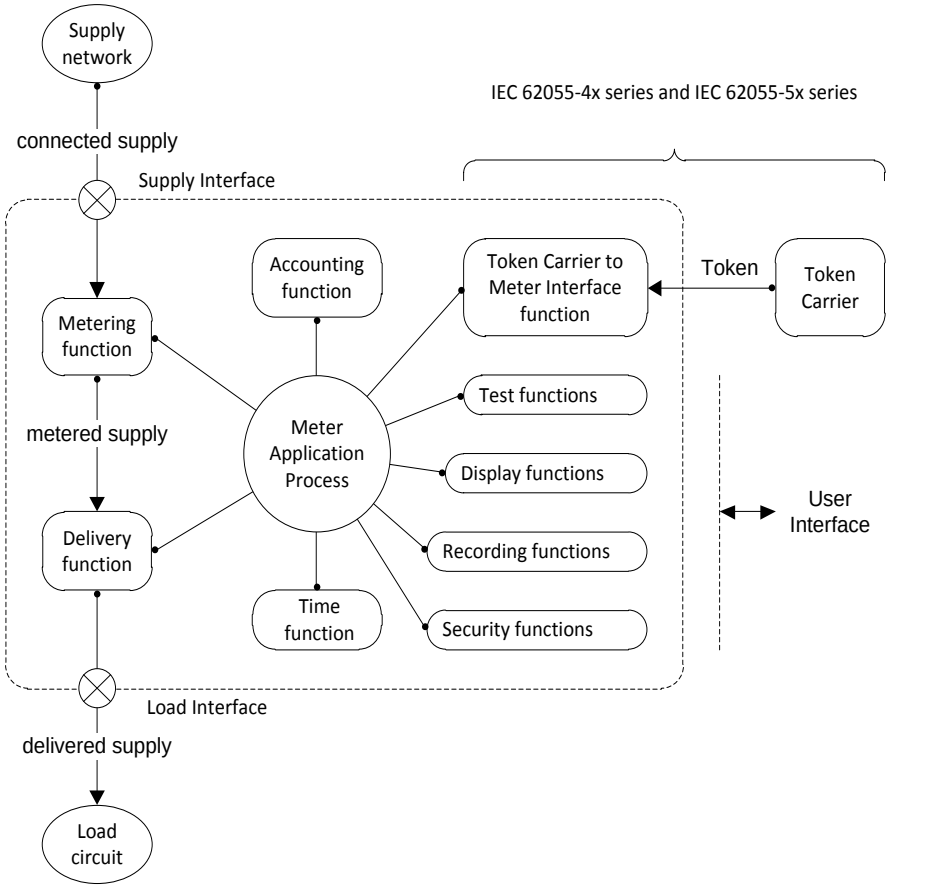
Les valeurs des chiffres binaires se situent dans la plage 0 à 1.

Les valeurs des chiffres décimaux se situent dans la plage 0 à 9.

Les valeurs des chiffres hexadécimaux se situent dans les plages 0 à 9, A à F et sont indiquées par “hex”.

5 Modèle de référence pour la spécification de transfert normalisé

5.1 Diagramme fonctionnel de référence pour compteur à paiement générique



Légende

Anglais	Français
Token	Jeton
Token Carrier	Support de jeton
Token Carrier to Meter Interface function	Fonction de l'interface Support de jeton/Compteur
Accounting function	Fonction de comptabilisation
Metering function	Fonction de comptage
Delivery function	Fonction de livraison
Supply network	Réseau d'approvisionnement
Load circuit	Circuit de charge
metered supply	alimentation comptée
delivered supply	alimentation livrée
connected supply	alimentation connectée
Display functions	Fonctions d'affichage
Recording functions	Fonctions d'enregistrement
Test functions	Fonctions d'essai
Security functions	Fonctions de sécurité
Time function	Fonctions de temps

Anglais	Français
Meter Application Process	Processus d'application de compteur
Supply Interface	Interface d'alimentation/approvisionnement
Load Interface	Interface de charge
User Interface	Interface utilisateur
IEC 62055-4x series and IEC 62055-5x series	Série IEC 62055-4x et série IEC 62055-5x

Figure 1 – Diagramme fonctionnel en blocs d'un compteur à paiement générique en une seule partie

La série IEC 62055-4x traite principalement du protocole de couche application et la série IEC 62055-5x du protocole de couche physique de la TokenCarrierToMeterInterface. Le TokenCarrier est inclus dans la Couche physique. Dans la présente Norme, le Decoder (décodeur, voir Article 3) est défini comme étant la partie intégrante du compteur à paiement où sont situées les fonctions de la Couche application de la TokenCarrierToMeterInterface et, donc, un DRN lui est alloué (voir 6.1.2.3).

NOTE Les MeterFunctionObjects font l'objet d'un débat plus approfondi en 5.5.

Dans un compteur à paiement en une seule partie, toutes les fonctions essentielles sont situées dans une seule enveloppe telle que représentée à la Figure 1 ci-dessus, cas dans lequel le décodeur est intégré à la fonction de comptage et le DRN peut donc être facultativement synonyme du numéro de série du compteur.

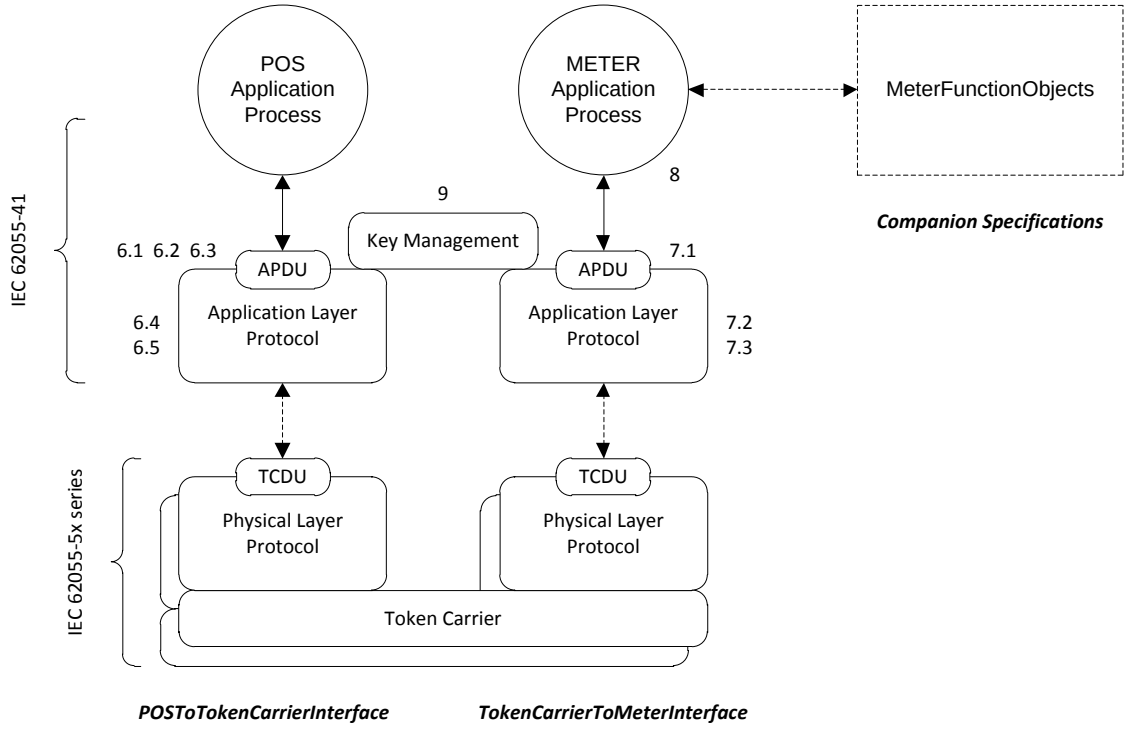
Dans un compteur à paiement en plusieurs parties, la TokenCarrierToMeterInterface peut être située dans une enveloppe séparée de celle de la fonction de comptage, par exemple, qui peut tout aussi bien se trouver dans un compteur autonome de son plein droit et ayant son propre numéro de série de compteur. Dans ce cas, le DRN n'est pas identique au numéro de série de compteur, mais est distinctement différent et est ainsi marqué sur l'enveloppe contenant le décodeur.

Dans tous les cas, il doit y avoir une seule mise en œuvre de la Couche application et il doit donc y avoir un seul DRN associé à un compteur à paiement, qu'il soit en une seule partie ou en plusieurs parties, même s'il peut y avoir plus d'une mise en œuvre de la Couche physique dans le même compteur à paiement.

Les fonctions de Couche application et les fonctions de Couche physique peuvent également être situées dans des enveloppes distinctes, auquel cas le marquage (voir 8.3) du DRN et du code EA est apposé sur la partie qui contient le point de connexion du TokenCarrier physique. Cela peut être un connecteur de câbles ou de modem pour un support de jeton virtuel, un clavier numérique pour un support de jeton numérique ou un lecteur de carte magnétique pour un support de jeton de carte magnétique, par exemple (voir également 5.2 pour plus d'exemples de supports de jeton).

Pour une description plus complète des classes de fonctions des compteurs à paiement, voir l'IEC 62055-21.

5.2 Modèle de référence de protocole STS



Légende

APDU ApplicationProtocolDataUnit; interface de données au protocole de couche application

TCDU TokenCarrierDataUnit; interface de données au protocole de couche physique

Les références des numéros d'articles pertinents dans la présente Norme sont indiquées à côté de chaque encadré.

Légende

Anglais	Français
Token Carrier	Support de jeton
Physical Layer Protocol	Protocole de couche physique
Application Layer Protocol	Protocole de couche application
TCDU	TCDU
METER Application Process	Processus d'application COMPTEUR
POS Application Process	Processus d'application POS
APDU	APDU
IEC 62055-5x series	Série IEC 62055-5x
Key Management	Gestion de clés
POSToTokenCarrierInterface	Interface POS/Support de jeton
Companion Specifications	Spécifications d'accompagnement
TokenCarrierToMeterInterface	Interface Support de jeton/Compteur
MeterFunctionObjects	Objets fonction de compteur

Figure 2 – STS modélisée comme une pile protocolaire OSI réduite à 2 couches

La STS est un protocole de transfert sécurisé de données entre un POS et un compteur à paiement utilisant un support de jeton comme support de transfert. Le protocole de couche application traite des jetons et des fonctions et processus de chiffrement, alors que le protocole de couche physique traite du codage réel des données du jeton sur un support de jeton (voir Figure 2).

Les exemples des dispositifs supports de jetons physiquement transportables sont: les numériques, les cartes magnétiques, les cartes à mémoire et les clés à mémoire. Des exemples de supports de jetons virtuels sont: modem RTPC, modem RNIS, modem GSM, modem GPRS, modem radio, modem PLC, connexions en infrarouge, connexions LAN et WAN, connexion locale directe Ils sont définis dans la série IEC 62055-5x.

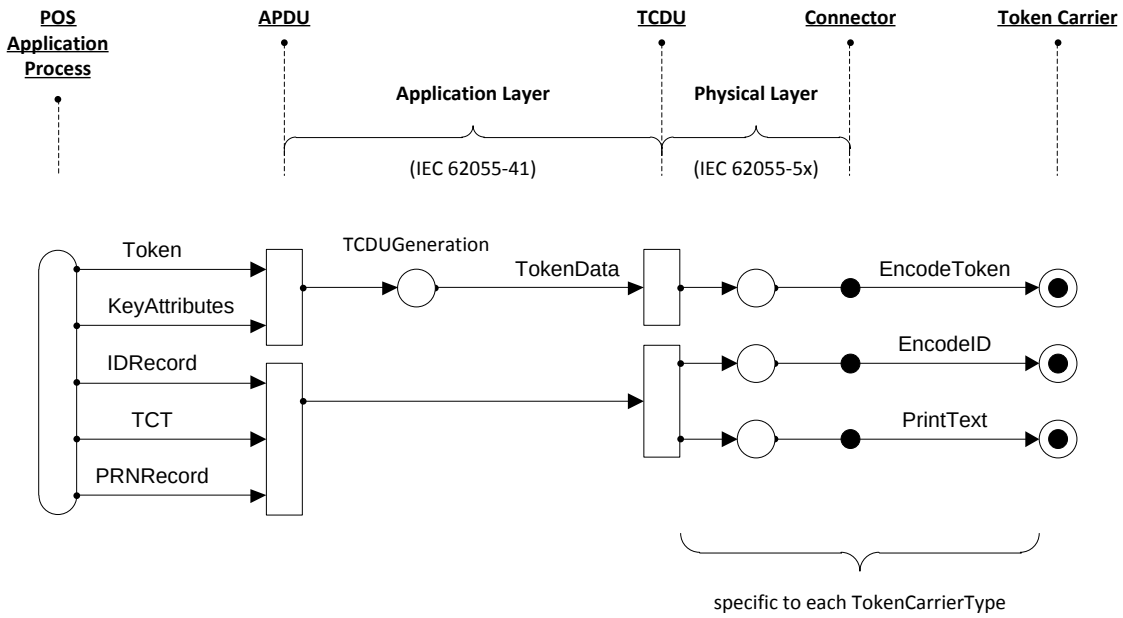
Il doit être noté que même si le modèle décrit principalement un POS au support de jeton au protocole de compteur à paiement, le même protocole est également applicable à n'importe quel autre dispositif qui exige de communiquer avec le compteur à paiement, par exemple CIS, SIG ou TSP portatif.

Bien qu'une architecture OSI réduite à 2 couches soit suivie dans la présente Norme, elle n'exclut pas une extension future pour inclure plus de couches si besoin est. Elle n'interdit pas non plus que le réalisateur intercale des couches supplémentaires entre les deux couches montrées dans le modèle.

L'APDU est l'interface de données au protocole de couche application, spécifiée dans l'IEC 62055-41, tandis que la TCDU est l'interface de données au protocole de couche physique, spécifiée dans la série IEC 62055-5x.

La STS dans la présente Norme définit un protocole de transfert de données en un seul sens (à savoir du POS vers le compteur à paiement), bien que le modèle de référence permette également un protocole de transfert dans les deux sens, qui peut être une exigence dans une future révision de la présente Norme.

5.3 Flux de données du POSApplicationProcess vers le TokenCarrier



IEC 0991/14

Légende

Anglais	Français
TCDUGeneration	Génération de TCDU
EncodeToken	Coder le jeton
specific to each TokenCarrierType	Spécifique à chaque type de support de jeton
POS Application Process	Processus application POS
APDU	APDU (Unité de données de processus application)
Application Layer (IEC 62055-41)	Couche application (IEC 62055-41)
TCDU	TCDU (Unité de données de support de jeton)
Physical Layer (IEC 62055-5x)	Couche physique (IEC 62055-5x)
Connector	Connecteur
Token Carrier	Support de jeton
Token	Jeton
TokenData	Données du jeton
KeyAttributes	Attributs de clé
IDRecord	Enregistrement d'ID
EncodeID	Coder l'ID
TCT	Type de support de jeton
PRNRecord	Enregistrement de PRN
PrintText	Imprimer le texte

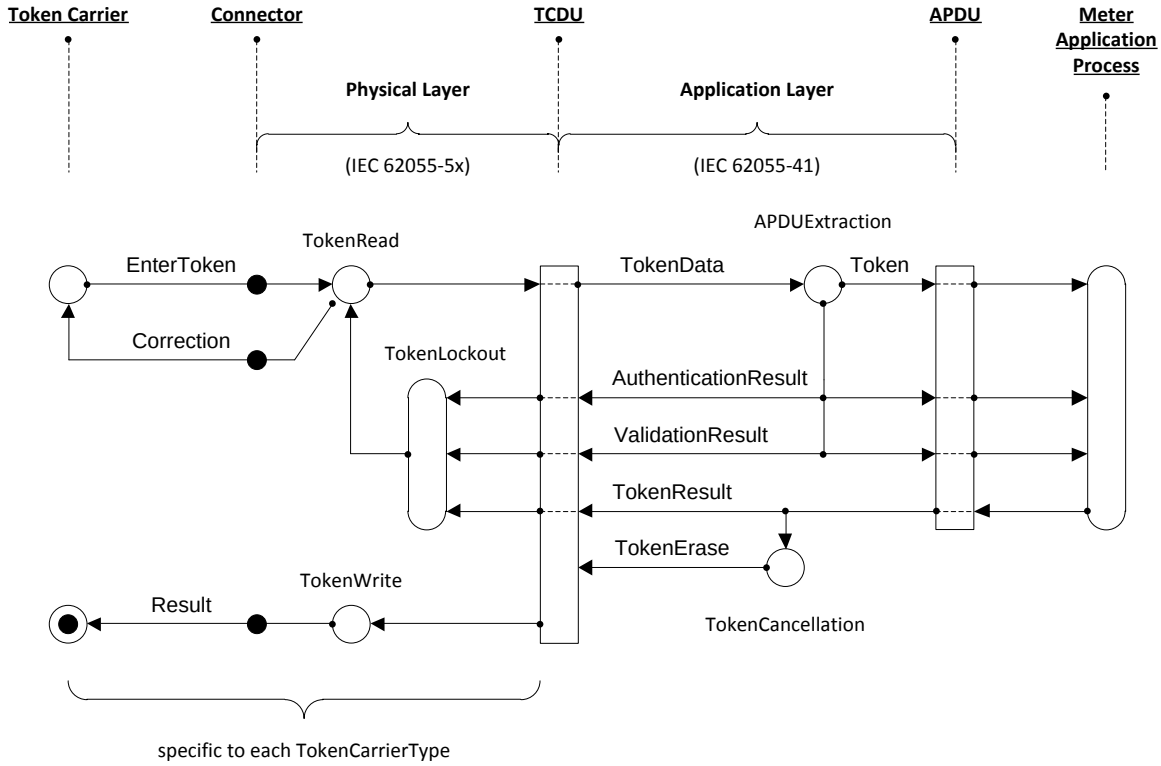
Figure 3 – Flux de données du POSApplicationProcess vers le TokenCarrier

Le flux de données du POSApplicationProcess vers le TokenCarrier est montré à la Figure 3.

Le POSApplicationProcess présente à l'APDU le jeton accompagné des KeyAttributes de la DecoderKey à utiliser pour chiffrer le jeton. Le protocole de couche application génère la DecoderKey, chiffre le jeton et présente les TokenData obtenues dans la TCDU. Le protocole de couche physique code les TokenData sur le TokenCarrier. En option, les données

d'identification du compteur à paiement peuvent également être codées sur le TokenCarrier (voir 5.2.4 dans l'IEC 62055-51:2007, par exemple) ainsi que le texte imprimé sur la surface extérieure (voir 5.1.5 dans l'IEC 62055-51:2007, par exemple). Cette partie du processus se termine essentiellement avec le codage des données sur le TokenCarrier, après quoi le TokenCarrier est transporté vers le compteur à paiement (habituellement par le client), où il est introduit dans le compteur à paiement par l'intermédiaire de la TokenCarrierInterface.

5.4 Flux de données du TokenCarrier vers le MeterApplicationProcess



IEC 0992/14

Légende

Anglais	Français
Token Carrier	Support de jeton
Connector	Connecteur
Physical Layer (IEC 62055-5x)	Couche physique (IEC 62055-5x)
TCDU	TCDU
Application Layer (IEC 62055-41)	Couche application (IEC 62055-41)
APDU	APDU
Meter Application Process	Processus application compteur
EnterToken	Introduire le jeton
TokenRead	Lire le jeton
Correction	Correction
TokenLockout	Verrouillage de jeton
APDUExtraction	Extraction d'unité APDU
TokenData	Données du jeton
Token	Jeton
AuthenticationResult	Résultat de l'authentification
ValidationResult	Résultat de la validation
TokenResult	Résultat jeton

Anglais	Français
TokenErase	Effacer le jeton
TokenCancellation	Annulation de jeton
TokenWrite	Écrire le jeton
Result	Résultat
specific to each TokenCarrierType	spécifique à chaque type de support de jeton

Figure 4 – Flux de données du TokenCarrier vers le MeterApplicationProcess

Le flux de données du TokenCarrier vers le MeterApplicationProcess est montré à la Figure 4.

Le processus d'introduction du jeton en provenance du TokenCarrier varie selon le TCT. De même, la nature du connecteur varie en fonction du TCT, dont un exemple peut être un clavier numérique ou un dispositif lecteur de carte magnétique prenant en charge les supports de jeton unidirectionnel tels que spécifiés dans l'IEC 62055-51.

NOTE D'autres types de connecteurs sont requis pour prendre en charge d'autres types de supports de jeton, tels qu'un dispositif de lecteur de clé à mémoire ou un connecteur enfichable à partir d'un terminal de saisie portable agissant comme support de jeton virtuel. De tels supports de jeton peuvent être spécifiés dans des parties supplémentaires de l'IEC 62055-5x dans le futur.

Le protocole de couche physique lit les données du jeton saisies et fournit une immédiate rétroaction corrective à l'utilisateur (voir 6.3 de l'IEC 62055-51:2007, par exemple). Les données du jeton saisies sont présentées dans la TCDU, d'où le protocole de couche application extrait le jeton par une opération appropriée de déchiffrement, validation et authentification, dont les résultats sont présentés au MeterApplicationProcess dans l'APDU. Après traitement et exécution de l'instruction à partir du jeton, le MeterApplicationProcess indique le résultat dans l'APDU pour que le protocole de couche application entreprenne une action ultérieure. Cela provoque normalement l'annulation du TID et la remise de l'instruction, par l'intermédiaire de la TCDU, au protocole de couche physique de parachever le processus de saisie de jeton par l'effacement des données du jeton (si cela est approprié) ou par l'écriture d'autres données pertinentes sur le TokenCarrier selon ce qui peut être approprié.

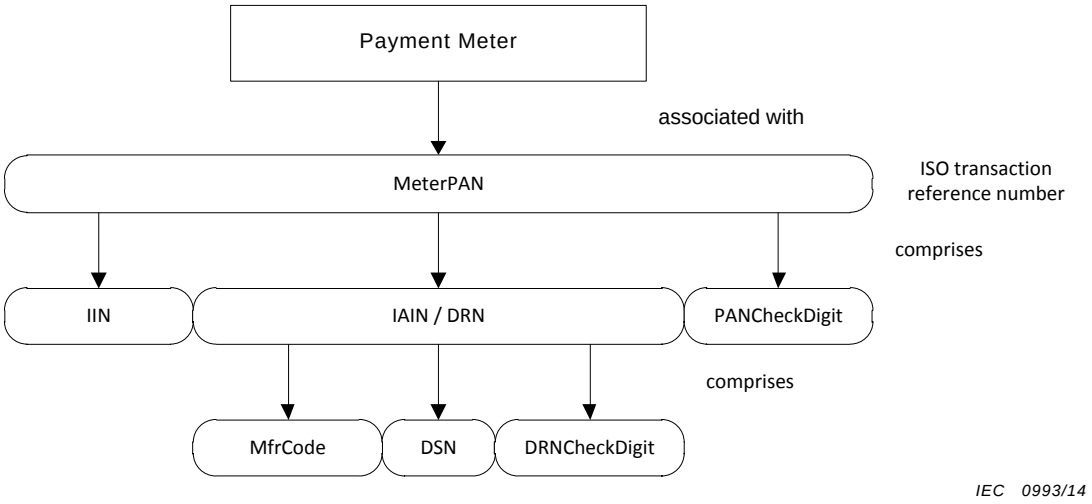
Pour certains types de TokenCarrier (un support de jeton virtuel à grande vitesse, par exemple), le protocole de couche physique peut utiliser une fonction de verrouillage de saisie de jeton pour protéger le compteur à paiement contre des tentatives de fraude. Typiquement, une telle fonction de verrouillage ralentit le débit effectif, auquel les jetons peuvent être introduits par l'interface du support de jeton particulier (voir 6.6.7 de l'IEC 62055-52:2008 par exemple).

5.5 MeterFunctionObjects / spécifications d'accompagnement

La Figure 1 permet de voir que la TokenCarrierToMeterInterface, qui inclut également le TokenCarrier, est traitée dans la série IEC 62055-4x et la série IEC 62055-5x. Les MeterFunctionObjects restants montrés dans le diagramme sont définis dans des spécifications d'accompagnement et ne sont pas normatifs dans la présente Norme.

Les spécifications d'accompagnement (voir Figure 2) sont sous le contrôle administratif (voir Article C.8) de la STS Association et servent à définir la fonctionnalité d'un compteur à paiement d'une manière normalisée, en utilisant une approche orientée objet.

5.6 Numéros de référence des transactions ISO



Légende

Anglais	Français
Payment Meter	Compteur à paiement
associated with	associé à
MeterPAN	Numéro de compte primaire du compteur
ISO transaction reference number	Numéro ISO de référence de la transaction
IIN	Numéro d'identification d'émetteur
IAIN / DRN	Numéro d'identification de compte individuel /Numéro de référence de décodeur
PANCheckDigit	Chiffre de vérification du numéro de compte primaire
comprises	comprend
MfrCode	Code constructeur
DSN	Numéro de série de décodeur
DRNCheckDigit	Chiffre de vérification du numéro de référence de décodeur

Figure 5 – Composition d'un numéro de référence de transaction ISO

Le numéro de référence de transaction ISO comprend les éléments de données et leurs relations tels que montrés à la Figure 5.

Une transaction à base de jeton (voir Article 3) constitue une activité financière qu'il est nécessaire de traiter conformément aux bonnes pratiques financières normalisées.

Le PrimaryAccountNumber (PAN) tel que défini par l'ISO/IEC 7812-1 sert à étiqueter des enregistrements, messages, demandes, autorisations et notifications de transaction, dans lesquels les parties prenantes de la transaction sont identifiables de façon univoque.

Un compteur à paiement est ainsi associé de façon unique à un MeterPAN, un numéro composé constitué de l'IIN et de l'IAIN/DRN, qui, à son tour, comprend le MfrCode et le DSN (voir 6.1.2).

6 Protocole de couche application POSToTokenCarrierInterface

6.1 APDU: ApplicationProtocolDataUnit

6.1.1 Éléments de données dans l'APDU

L'APDU, qui est l'interface de données entre le POSApplicationProcess et le protocole de couche application, comprend les éléments de données indiqués dans le Tableau 1.

Tableau 1 – Éléments de données dans l'APDU

Élément	Contexte	Format	Référence
MeterPAN	MeterPrimaryAccountNumber pour l'identification conforme à l'ISO pour le compteur à paiement	18 chiffres	6.1.2
TCT	Désigne quel TokenCarrierType il convient d'utiliser dans le protocole de couche physique pour transporter le jeton vers le compteur à paiement	2 chiffres	6.1.3
DKGA	Désigne quel DecoderKeyGenerationAlgorithm est à utiliser pour générer la DecoderKey	2 chiffres	6.1.4
EA	Désigne quel algorithme de chiffrement est à utiliser pour chiffrer les données du jeton	2 chiffres	6.1.5
SGC	Désigne le SupplyGroupCode auquel le compteur à paiement est alloué	6 chiffres	6.1.6
TI	Désigne le TariffIndex auquel le compteur à paiement est relié	2 chiffres	6.1.7
KRN	Désigne le KeyRevisionNumber sur lequel la DecoderKey se trouve	1 chiffre	6.1.8
KT	Désigne le KeyType sur lequel la DecoderKey se trouve	1 chiffre	6.1.9
KeyExpiryNumber	Nombre associé à la VendingKey et à une DecoderKey qui détermine la durée pendant laquelle la clé reste valide	8 bits	6.1.10
Token	Les données du jeton réelles à transférer au compteur à paiement avant chiffrement et traitement	66 bits	6.2.1
IDRecord	Données d'identification facultatives destinées à être codées sur la carte d'identification d'un compteur à paiement ou sur un support de jeton avec le jeton	35 chiffres	Tableau 2
PRNRecord	Des données d'impression facultatives destinées à être imprimées en même temps que le codage du jeton sur le TokenCarrier. Certains supports de jeton tels que les dispositifs de carte magnétique sur papier permettent l'impression sur la surface de la carte elle-même et cette opération peut être intégrée avec le dispositif de codage de carte magnétique. Le contenu et le format ne sont pas spécifiés et chaque système peut les définir à sa discrétion selon ses exigences particulières.	Texte non défini	x

L'IDRecord facultatif comprend les éléments de données consignés dans le Tableau 2.

Tableau 2 – Éléments de données dans l'IDRecord

Élément	Contexte	Format	Référence
MeterPAN	MeterPrimaryAccountNumber pour l'identification conforme à l'ISO pour le compteur à paiement	18 chiffres	6.1.2
DOE	Date d'expiration facultative des données d'identification telles que codées sur la carte d'identification d'un compteur à paiement ou le support de jeton (voir l'IEC 62055-51 pour un exemple)	4 chiffres	6.1.11
TCT	Indique quel TokenCarrierType est associé à ce MeterPAN	2 chiffres	6.1.3
EA	Indique quel algorithme de chiffrement est associé à ce MeterPAN	2 chiffres	6.1.5
SGC	Indique quel SupplyGroupCode est associé à ce MeterPAN	6 chiffres	6.1.6
TI	Indique quel TariffIndex est associé à ce MeterPAN	2 chiffres	6.1.7
KRN	Indique quel KeyRevisionNumber est associé à ce MeterPAN	1 chiffre	6.1.8

6.1.2 MeterPAN: MeterPrimaryAccountNumber

6.1.2.1 Éléments de données dans le MeterPAN

Le MeterPAN est un numéro d'identification unique pour chaque compteur à paiement conforme à la STS. Il comporte les 3 parties données dans le Tableau 3 et il est conforme à la définition du PAN (PrimaryAccountNumber) de l'ISO/IEC 7812-1.

Tableau 3 – Éléments de données dans le MeterPAN

Élément	Contexte	Format	Référence
IIN	IssuerIdentificationNumber (Numéro d'identification d'émetteur)	4/6 chiffres	6.1.2.2
IAIN / DRN	IndividualAccountIdentificationNumber / DecoderReferenceNumber	11/13 chiffres	6.1.2.3
PANCheckDigit	Formule pour vérifier l'intégrité de l'IIN et de l'IAIN	1 chiffre	6.1.2.4
NOTE Le premier chiffre de l'IIN est le chiffre de poids fort du MeterPAN à 18 chiffres et le PANCheckDigit est le chiffre de poids faible.			

Voir aussi l'Annexe C pour le Code de bonnes pratiques de gestion de cet élément de données.

6.1.2.2 IIN: IssuerIdentificationNumber

L'IIN est un nombre unique de 6 chiffres qui définit un domaine, sous lequel d'autres valeurs d'IAIN (c'est-à-dire des valeurs de DRN) peuvent être émises pour être utilisées au sein du domaine défini.

L'intention et le but initiaux de l'IIN étaient de pouvoir étiqueter des transactions financières afin de les identifier de façon unique et les acheminer vers les comptes financiers appropriés engagés dans une transaction. L'IIN était donc censé être émis par l'ISO dans le cadre du plan d'enregistrement donné dans l'ISO/IEC 7812-1 et l'ISO/IEC 7812-2. Cependant, cela s'est avéré irréalisable dans la pratique et la valeur 600727 pour l'IIN est dès lors devenue la norme *de facto* pour les systèmes hérités utilisant un DRN de 11 chiffres.

Par la suite, il est devenu nécessaire de prendre également des dispositions pour les DRN de 13 chiffres (tels que définis en 6.1.2.3.1) et dans ce cas, l'IIN doit être 0000 (quatre zéros).

Voir aussi C.3.2 pour la gestion de cet élément de données.

6.1.2.3 IAIN: IndividualAccountIdentificationNumber/ DRN: DecoderReferenceNumber

6.1.2.3.1 Éléments de données dans l'IAIN / DRN

Un DRN unique doit être alloué au dispositif qui exécute le protocole de couche application dans un compteur à paiement conforme à la STS.

NOTE Dans un grand nombre de systèmes, la partie "décodeur" est intégrée à la partie de comptage et, donc, le DRN peut être synonyme du numéro de série de compteur.

Il s'agit d'un nombre de 11/13 chiffres constitué des éléments de données indiqués dans le Tableau 4.

Tableau 4 – Éléments de données dans l'IAIN / DRN

Élément	Contexte	Format	Référence
MfrCode	Nombre pour identifier de façon unique un constructeur de compteur à paiement	2/4 chiffres	6.1.2.3.2
DSN	Numéro de série à huit chiffres alloué par le constructeur	8 chiffres	6.1.2.3.3
DRNCheckDigit	Check Digit; Formule pour vérifier l'intégrité du MfrCode et du DSN	1 chiffre	6.1.2.3.4
NOTE Le MfrCode est constitué des 2/4 chiffres de poids fort du DRN de 11/13 chiffres et le DRNCheckDigit est le chiffre de poids faible.			

Les valeurs du MfrCode doivent toujours être justifiées à droite et complétées de 0 à gauche.

Le DSN doit être justifié à droite et complété de 0 à gauche pour obtenir une chaîne complète de 8 chiffres.

6.1.2.3.2 MfrCode: ManufacturerCode

Le MfrCode est un nombre de 2/4 chiffres qui doit être utilisé pour identifier de façon unique le constructeur du compteur à paiement.

La STS Association fournit un service pour l'allocation des valeurs de MfrCode pour identifier de façon unique les constructeurs afin d'assurer l'interopérabilité des matériels conformes à la STS.

Voir aussi C.3.3 pour la gestion de cet élément de données.

6.1.2.3.3 DSN: DecoderSerialNumber

Le DSN est un numéro de série unique à 8 chiffres qui est généré en interne par le constructeur. Chaque constructeur est responsable de l'unicité du DSN en ce qui concerne son MfrCode.

Voir aussi C.3.4 pour la gestion de cet élément de données.

6.1.2.3.4 DRNCheckDigit

Le DRNCheckDigit est un chiffre unique utilisé pour valider l'intégrité des valeurs du MfrCode et de DSN lorsqu'elles sont saisies manuellement ou lues par une machine. Il s'agit d'un chiffre de vérification modulo 10, calculé à l'aide de la formule de Luhn, de la façon illustrée dans l'Annexe B de l'ISO/IEC 7812-1:2000. Il est calculé sur les 10/12 chiffres précédents du DRN généré par la concaténation des valeurs du MfrCode et du DSN.

6.1.2.4 PANCheckDigit

Le PANCheckDigit est un chiffre unique utilisé pour valider l'intégrité des valeurs de l'IIN et de l'IAIN lorsqu'elles sont saisies manuellement ou lues par une machine. La méthode utilisée pour calculer la valeur de PANCheckDigit est donnée en 4.4 de l'ISO/IEC 7812-1:2000. Cette valeur est calculée sur les 17 chiffres précédents du MeterPAN généré par la concaténation des valeurs de l'IIN et de l'IAIN.

6.1.3 TCT: TokenCarrierType

Il s'agit d'un nombre de 2 chiffres utilisé pour identifier de façon unique le type du support de jeton sur lequel il convient de coder le jeton pour le transfert vers le compteur à paiement. Les valeurs pour les types de support de jeton sont données dans le Tableau 5.

Tableau 5 – Types de support de jeton

Code	TokenCarrier	Commentaires
00	Réservé	Pour affectation future
01	Carte magnétique	Conformément à l'IEC 62055-51
02	Numérique	Conformément à l'IEC 62055-51
03-06	Réservé	Systèmes hérités utilisant des technologies de support de jeton propriétaires
07	Support de jeton virtuel (Virtual Token Carrier (VTC07))	Conformément à l'IEC 62055-52
08-99	Réservé	Pour affectation future

Les valeurs inférieures à 10 doivent être justifiées à droite et complétées de 0 à gauche (par exemple: 01, 02-09).

6.1.4 DKGA: DecoderKeyGenerationAlgorithm

Il s'agit d'un nombre de 2 chiffres utilisé pour identifier de façon unique l'algorithme à utiliser pour générer la DecoderKey. Les valeurs des codes de DKGA sont données dans le Tableau 6.

Tableau 6 – Codes de DKGA

Code	Algorithme DKG	Commentaires	Référence
00	Réservé	Pour affectation future	x
01	DKGA01	Nombre limité des premiers compteurs à paiement hérités conformes à la STS. Annulé et remplacé par DKGA02	6.5.3.3
02	DKGA02	Système utilisant la diversification de VendingKey en DES 64 bits	6.5.3.4
03	DKGA03	Système utilisant la double diversification de VendingKey en DES 64 bits	6.5.3.5
04-99	Réservé	Pour affectation future	X
DKGA02 est l'algorithme à utiliser pour les systèmes actuels, soumis aux critères pour DKGA01. DKGA03 est l'algorithme à utiliser pour les futurs systèmes exigeant un plus haut niveau de sécurité en ce qui concerne la protection de la VendingKey par attaque par force brute. Il convient que l'introduction de DKGA03 coïncide préférentiellement avec le passage de STA à DEA (du code EA 07 au code EA 09). Voir aussi 6.1.5.			

Les valeurs inférieures à 10 doivent être justifiées à droite et complétées de 0 à gauche (par exemple: 01, 02-09).

6.1.5 EA: EncryptionAlgorithm

Il s'agit d'un nombre de 2 chiffres utilisé pour identifier de façon unique l'algorithme à utiliser pour chiffrer les données du jeton. Les valeurs des codes EA sont données dans le Tableau 7.

Tableau 7 – Codes EA

Code	EncryptionAlgorithm	Commentaires	Référence
00	Réservé	Pour affectation future	x
01-06	Réservé	Systèmes propriétaires hérités	x
07	STA	Systèmes utilisant l'algorithme de transfert normalisé tel que défini dans la présente Norme	6.5.4.1
08	Réservé	Systèmes propriétaires hérités	x
09	DEA	Systèmes utilisant l'algorithme de chiffrement de données tel que défini dans l'ANSI X3.92	6.5.5
10	Réservé	Systèmes propriétaires hérités	x
11-99	Réservé	Pour affectation future	x
Il est recommandé de coordonner le choix du code EA 09 avec le choix de DKGA03 afin de réduire au maximum l'effet sur les systèmes existants dans la base installée (voir 6.1.4).			

Les valeurs inférieures à 10 doivent être justifiées à droite et complétées de 0 à gauche. Par exemple: 01, 02-09.

6.1.6 SGC: SupplyGroupCode

Il s'agit d'un nombre unique de 6 chiffres alloué à l'entreprise de distribution, qui est enregistré au sein du KMS. Il est utilisé pour identifier de façon unique un sous-groupe de compteurs à paiement au sein du domaine de fourniture ou de distribution de l'entreprise de distribution. Chaque SupplyGroup a une VendingKey qui lui est associée et, donc, chaque compteur à paiement dans le SupplyGroup a une DecoderKey dérivée qui lui est associée. L'autorisation des ventes de jetons est donc commandée par la distribution sélective de telles clés VendingKey et de tels codes SGC à des agents de vente de jetons habilités exploitant des services POS pour le compte d'entreprises de distribution.

La gestion des SGC et la gestion des VendingKey sont complètement sous le contrôle du KMS et sont soumises à un tel Code de bonnes pratiques.

Les valeurs inférieures à 6 chiffres décimaux doivent être justifiées à droite et complétées de 0 à gauche. Par exemple: 000001, 000002..000009.

Le SGC hérite de son type de l'attribut KT de la VendingKey (voir 6.5.2.2.1), à laquelle il est associé selon le Tableau 8.

Tableau 8 – Types de SGC et types de clés

KT	Type de SGC	Type de VendingKey (voir 6.5.2.2.1)	Type de DecoderKey (voir 6.5.2.3.1)
0	Initialisation	Non spécifié	DITK
1	Default (valeur par défaut)	VDDK	DDTK
2	Unique	VUDK	DUTK
3	Common (commune)	VCDK	DCTK

Voir aussi C.2.2 pour le Code de bonnes pratiques de gestion de cet élément de données.

6.1.7 TI: TariffIndex

Nombre de 2 chiffres associé à un tarif particulier qui est alloué à un consommateur particulier. La maintenance et le contenu des tableaux de tarif sont de la responsabilité de l'entreprise de distribution.

Les valeurs inférieures à 10 doivent être justifiées à droite et complétées d'un 0 à gauche (par exemple: 01, 02.. 09).

Le TI est également codé dans la DecoderKey, ce qui signifie que quand un client passe d'un TI à l'autre, il faut aussi que sa DecoderKey change (voir 6.5.2.1).

NOTE Le codage de cette valeur lorsqu'elle est utilisée dans le ControlBlock pour la génération de clé de décodeur (voir 6.5.3.2) est sous la forme de deux chiffres hexadécimaux, alors que le codage tel qu'utilisé dans le jeton Set2ndSectionDecoderKey (voir 6.2.8) est sous la forme d'un nombre binaire de 8 bits. Dans ces cas, un index de tarif de 99 en décimal est codé en une chaîne binaire, respectivement 10011001 et 0110 0011.

Voir aussi l'Article C.9 pour le Code de bonnes pratiques de gestion de cet élément de données.

6.1.8 KRN: KeyRevisionNumber

Il s'agit d'un nombre de 1 chiffre dans la plage 1 à 9, qui est associé à une version de la VendingKey et à la DecoderKey correspondante.

Voir 6.5.2.5 pour une définition détaillée de cet élément de données.

6.1.9 KT: KeyType

Il s'agit d'un nombre de 1 chiffre dans la plage 0 à 3 associé à une propriété de la VendingKey et, donc aussi, à la DecoderKey correspondante, qui est dérivée de la VendingKey.

Voir 6.5.2 pour une définition détaillée de cet élément de données.

6.1.10 KEN: KeyExpiryNumber

Un KEN est associé à chaque VendingKey par le KMS et définit l'instant où une VendingKey et toute DecoderKey correspondante expireront, après quoi il devient non valide pour une utilisation future, moyennant certaines concessions.

Le KEN correspond aux 8 bits de poids fort du TID de 24 bits. Aucun identificateur de jeton dont les 8 bits de poids fort sont supérieurs au KEN d'une clé donnée ne peut être chiffré ou déchiffré avec la clé en question.

Voir 6.5.2.6 pour une définition détaillée de cet élément de données.

Voir aussi C.2.4 pour le Code de bonnes pratiques de gestion de cet élément de données.

6.1.11 DOE: DateOfExpiry

L'utilisation de cette date est facultative et elle est associée à une période de validité pour les données relatives à l'identité qui sont codées sur un dispositif support d'identité. Par exemple: une carte d'identification de compteur à paiement ou un second enregistrement codé sur le TokenCarrier avec les données du jeton. Dans certaines mises en œuvre, il s'est avéré utile de laisser le consommateur rapporter un support de jeton utilisé pour qu'il soit son

identification de décodeur au POS lorsqu'il achète son prochain jeton. (Voir 5.1.4 et 5.2.4.9 de l'IEC 62055-51:2007, par exemple).

Cette date peut également être utilisée, par exemple, dans les cas où un tarif concessionnaire a été accordé à un client pendant une durée limitée. La date codée est le dernier mois pendant lequel la carte est valide.

DOE est au format AAMM et doit toujours contenir 4 chiffres.

Lorsque AA ou MM est inférieur à 10, il doit être justifié à droite et complété d'un 0 à gauche (par exemple: 01, 02, 09, etc.).

Lorsque la DOE dans l'IDRecord n'est pas utilisée, alors AAMM = 0000.

Les valeurs du code de DOE pour l'année et le mois sont données dans le Tableau 9 et le Tableau 10.

Tableau 9 – Codes de DOE pour l'année

AA	Représente
00	2000 ou alors la DOE n'est pas utilisée (voir aussi le Tableau 10)
01 – 99	2001 – 2099

Tableau 10 – Codes de DOE pour le mois

MM	Représente
00	La DOE n'est pas utilisée (voir aussi le Tableau 9)
01 – 12	janvier – décembre
13 – 99	Non valide

6.2 Jetons

6.2.1 Format de définition de jeton

L'élément TokenData dans l'APDU est un nombre binaire de 66 bits constitué de plusieurs champs d'éléments de données plus petits, selon lesquels divers procédés sont initiés dans le MeterApplicationProcess et divers bits d'informations sont transférés aux registres du compteur à paiement.

Le format de définition pour les jetons de 6.2.2 à 6.2.14 est donné dans le Tableau 11.

Tableau 11 – Format de définition de jeton

Nom d'élément de données	Exemple:	Class (c'est-à-dire: Classe), SubClass (c'est-à-dire: Sous-classe), RND, TID, Amount (c'est-à-dire: Montant), CRC, etc.
Nombre de bits	Exemple:	2 bits, 4 bits, 24 bits, 16 bits, etc.
Plage de valeurs	Exemple:	1, 2, 5-15, etc.

6.2.2 Classe 0: TransferCredit

Class	SubClass	RND	TID	Amount	CRC
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
0	0 = électricité 1 = eau 2 = gaz Réservé: 3 = temps 4 = monnaie 5-15 = affectation future				
NOTE Les valeurs 3-4 de SubClass sont réservées par la STS Association pour les applications autres que l'électricité, le gaz et l'eau, les valeurs 5-15 étant réservées pour une affectation future.					

Action: Transférer le crédit au compteur à paiement à la valeur définie par le champ Amount et pour le type de service défini dans le champ SubClass.

6.2.3 Classe 1: InitiateMeterTest/Display

Class	SubClass	Control (c'est-à-dire: Contrôle)	MfrCode	CRC
2 bits	**4 bits**	**36/28 bits**	**8/16 bits**	**16 bits**
1	0 = définie par STS	Le contrôle de position de bits du numéro d'essai/affichage pour les codes de constructeur de 2 chiffres. Utiliser 36 bits.	0 (8 bits)	
1	1 = définie par STS	Le contrôle de position de bits du numéro d'essai/affichage pour les codes de constructeur de 4 chiffres. Utiliser 28 bits.	0 (16 bits)	
1	2-5 = réservées pour une affectation future.	Réservé pour une affectation future.	Réservé pour une affectation future.	
1	6-10 = utilisation propriétaire.	Pour les codes de constructeur de 4 chiffres. Si pas utilisé, mettre à zéro (28 bits)	0100-9999 (16 bits)	
1	11-15 = utilisation propriétaire.	Pour les codes de constructeur de 2 chiffres. Si pas utilisé, mettre à zéro (36 bits)	00-99 (8 bits)	

Action: Initier la fonction d'essai ou d'affichage dans le compteur à paiement en fonction du profil binaire défini dans le champ Control.

6.2.4 Classe 2: SetMaximumPowerLimit

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

Action: Charger le registre de limite de puissance maximale dans le compteur à paiement avec la valeur donnée dans le champ MPL.

6.2.5 Classe 2: ClearCredit

Class	SubClass	RND	TID	Register (c'est-à-dire: Registre)	CRC
2 bits	4 bits	4 bits	24 bits	16 bits	16 bits
2	1				

Action: Vider le registre de crédit correspondant (tel qu'indiqué par le champ Register) dans le compteur à paiement à zéro.

6.2.6 Classe 2: SetTariffRate

Class	SubClass	RND	TID	Rate (c'est-à-dire: Taux)	CRC
2 bits	4 bits	4 bits	24 bits	16 bits	16 bits
2	2				

Action: Charger le registre tarifaire dans le compteur à paiement avec la valeur donnée dans le champ Rate.

Ce jeton est réservé par la STS Association pour une définition future.

6.2.7 Classe 2: Set1stSectionDecoderKey

Class	SubClass	KENHO	KRN	RO	Res (c'est-à-dire: Réservé)	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	x	0-3		

Action: Charger le DecoderKeyRegister avec la 1ère moitié de la nouvelle DecoderKey, sous réserve d'un chargement authentique d'un jeton Set2ndSectionDecoderKey.

6.2.8 Classe 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: Charger le DecoderKeyRegister avec la 2ème moitié de la nouvelle DecoderKey, sous réserve d'un chargement authentique d'un jeton Set1stSectionDecoderKey.

6.2.9 Classe 2: ClearTamperCondition

Class	SubClass	RND	TID	Pad (c'est-à-dire: Bourrage)	CRC
2 bits	4 bits	4 bits	24 bits	16 bits	16 bits
2	5			0	

Action: Vider le registre de statut de fraude dans le compteur à paiement et annuler tous les processus de contrôle résultants qui peuvent être en cours.

6.2.10 Classe 2: SetMaximumPhasePowerUnbalanceLimit

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

Action: Charger le registre de limite de déséquilibre maximal des phases dans le compteur à paiement avec la valeur donnée dans le champ MPPUL. Voir aussi 8.12 pour plus de détails sur l'action de cette fonction dans le compteur à paiement.

6.2.11 Classe 2: SetWaterMeterFactor

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

Action: Charger le registre du facteur de compteur d'eau dans le compteur à paiement avec la valeur donnée dans le champ WMFactor.

Ce jeton est réservé par la STS Association pour les applications relatives à l'eau.

6.2.12 Classe 2: Réservée pour l'usage selon la STS

Class	SubClass	RND	TID	ResData (c'est-à-dire: Données réservées)	CRC
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
2	8-10				

Action: Réservée pour une définition future par la STS Association.

Cette plage de jetons est réservée par la STS Association pour une affectation future.

6.2.13 Classe 2: Réservée pour un usage propriétaire

Class	SubClass	RND	TID	PropData (c'est-à-dire: Données propriétaires)	CRC
2 bits	**4 bits**	**4 bits**	**24 bits**	**16 bits**	**16 bits**
2	11-15				

Action: Définie par le constructeur.

Cette plage de jetons est réservée pour une définition et un usage propriétaires.

La présente Norme ne fournit de protection contre la collision entre les usages de constructeurs de cet espace de jetons. La génération et le contrôle de ces jetons doivent donc toujours être sous la gestion directe du constructeur approprié et ne doivent jamais être disponibles sur les systèmes de distribution automatique pour usage général au sein des systèmes de comptage de paiement conformes à la STS.

6.2.14 Classe 3: Réservée pour l'usage selon la STS

Class	SubClass	Res (c'est-à-dire: Réservé)
2 bits	**4 bits**	**60 bits**
3	0-15	

Action: Réservée pour une définition future par la STS Association.

Cette plage de jetons est réservée par la STS Association pour une affectation future.

6.3 Éléments de données du jeton

6.3.1 Éléments de données utilisés dans des jetons

Les éléments de données consignés dans le Tableau 12 sont utilisés dans les jetons dans diverses combinaisons.

Tableau 12 – Éléments de données utilisés dans des jetons

Élément	Nom	Format	Référence
Amount	TransferAmount (voir aussi 6.2.2)	16 bits	6.3.6
Class	TokenClass (voir aussi 6.2.2 à 6.2.14)	2 bits	6.3.2
Control	InitiateMeterTest/DisplayControlField (voir aussi 6.2.3)	36/28 bits	6.3.8
CRC	CyclicRedundancyCode (voir 6.2.2 à 6.2.13)	16 bits	6.3.7
KENHO	KeyExpiryNumberHighOrder (voir aussi 6.2.7)	4 bits	6.3.16
KENLO	KeyExpiryNumberLowOrder (voir aussi 6.2.8)	4 bits	6.3.17
KRN	KeyRevisionNumber (voir aussi 6.2.7)	4 bits	6.1.8
KT	KeyType (voir aussi 6.2.7)	2 bits	6.1.9
MfrCode	ManufacturerCode (voir aussi 6.2.3)	8/16 bits	6.1.2.3.2
MPL	MaximumPowerLimit (voir aussi 6.2.4)	16 bits	6.3.9
MPPUL	MaximumPhasePowerUnbalanceLimit (voir aussi 6.2.10)	16 bits	6.3.10
NKHO	NewKeyHighOrder (voir aussi 6.2.7)	32 bits	6.3.14
NKLO	NewKeyLowOrder (voir aussi 6.2.8)	32 bits	6.3.15
Pad	Compléter la valeur avec des 0 (voir aussi 6.2.9)	16 bits	x
PropData	Champ de données propriétaires (voir aussi 6.2.13)	16 bits	x
Rate	[TariffRate] Pour une définition future (voir aussi 6.2.6)	16 bits	6.3.11
Register	RegisterToClear (voir aussi 6.2.5)	16 bits	6.3.13
Res	Réservé pour une affectation future (voir aussi 6.2.7 et 6.2.14)	1 bit	x
ResData	Champ de données réservées pour une affectation future (voir aussi 6.2.12)	16 bits	x
RND	RandomNumber (voir aussi 6.2.2 à 6.2.13)	4 bits	6.3.4
RO	RolloverKeyChange (voir aussi 6.2.7)	1 bit	6.3.18
SubClass	TokenSubClass (voir aussi 6.2.2 à 6.2.14)	4 bits	6.3.3
TI	TariffIndex (voir aussi 6.2.8)	8 bits	6.1.7
TID	TokenIdentifier (voir aussi 6.2.2 à 6.2.14)	24 bits	6.3.5.1
WMFactor	[WaterMeterFactor] Réservé par la STS Association pour une application relative à l'eau (voir aussi 6.2.11)	16 bits	6.3.12

6.3.2 Classe: TokenClass

Les jetons sont classés en 4 principaux domaines fonctionnels tels que donnés dans le Tableau 13.

Tableau 13 – Classes de jetons

TokenClass	Fonction
0	Transfert de crédit
1	Gestion non spécifique à un compteur
2	Gestion spécifique à un compteur
3	Réservé pour une affectation future

Les jetons de Classe 0 et Classe 2 sont chiffrés en utilisant la DecoderKey, alors que les jetons de Classe 1 ne sont pas chiffrés et peuvent donc être utilisés par tout compteur à paiement conforme à la STS.

6.3.3 SubClass: TokenSubClass

Une sous-classification plus poussée de la TokenClass est donnée dans le Tableau 14.

Tableau 14 – Sous-classes de jetons

<table>
<tr><th rowspan="2">Token SubClass</th><th colspan="4">TokenClass</th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>0</td><td>TransferCredit (électricité)</td><td>InitiateMeterTest/ Display pour le MfrCode de 2 chiffres</td><td>SetMaximumPowerLimit</td><td rowspan="16">Réservé par la STS Association pour une affectation future</td></tr>
<tr><td>1</td><td>TransferCredit (eau)</td><td>InitiateMeterTest/ Display pour le MfrCode de 4 chiffres</td><td>ClearCredit</td></tr>
<tr><td>2</td><td>TransferCredit (gaz)</td><td rowspan="5">Réservé par la STS Association pour une affectation future</td><td>SetTariffRate</td></tr>
<tr><td>3</td><td>TransferCredit (temps)
Réservé par la STS Association pour des applications relatives au temps de connexion</td><td>Set1stSectionDecoderKey</td></tr>
<tr><td>4</td><td>TransferCredit (monnaie)
Réservé par la STS Association pour des applications relatives à la monnaie</td><td>Set2ndSectionDecoderKey</td></tr>
<tr><td>5</td><td rowspan="11">Réservé par la STS Association pour une affectation future</td><td>ClearTamperCondition</td></tr>
<tr><td>6</td><td rowspan="4">Réservée pour un usage propriétaire pour le MfrCode de 4 chiffres</td><td>SetMaximumPhasePowerUnbalanceLimit</td></tr>
<tr><td>7</td><td>SetWaterMeterFactor
Réservé par la STS Association pour des applications relatives à l'eau</td></tr>
<tr><td>8</td><td rowspan="3">Réservé par la STS Association pour une affectation future</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td><td rowspan="5">Réservée pour un usage propriétaire pour le MfrCode de 2 chiffres</td><td rowspan="5">Réservée pour un usage propriétaire</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

La génération de ce nombre de 4 bits est un instantané des quatre bits de poids faible d'au moins un compteur de millisecondes. L'inclusion d'un nombre aléatoire dans les données à transférer renforce la sécurité du transfert de jeton en assurant, avec une probabilité de 16:1, que deux jetons quelconques contenant des données identiques à transférer n'auront pas le même profil binaire.

6.3.5 TID: TokenIdentifier

6.3.5.1 Calcul de TID

Le champ TID est dérivé de la date et de l'heure d'émission et indique le nombre de minutes écoulées d'une date et d'une heure de référence STS. Ce champ est une représentation binaire 24 bits des minutes écoulées.

Afin de prendre en compte le fait que le TID repasse par zéro tous les 31 ans, trois dates de référence STS sont définies, à savoir:

- 01 janvier 1993, 00:00:00
- 01 janvier 2014, 00:00:00
- 01 janvier 2035, 00:00:00

Avec un format de date et heure de AAAA:MM:JJ:hh:mm:ss, l'ensemble date et heure de référence de la STS "1993:01:01:00:00:00" correspond à TID de 0.

Le calcul de minutes écoulées doit prendre en compte les années bissextiles.

La règle pour déterminer une année bissextile est la suivante:

- le mois de février doit avoir un jour supplémentaire dans toutes les années qui sont également divisibles par 4, excepté les années de siècle (celles qui finissent par 00), qui reçoivent le jour supplémentaire seulement si elles sont également divisibles par 400. Ainsi, 1996 était une année bissextile alors que 1999 ne l'était pas. Les années 1600, 2000 et 2400 sont des années bissextiles, mais 1700, 1800, 1900 et 2100 ne le sont pas.

Dans la représentation binaire du TID, le bit le plus à gauche représente le bit de poids fort.

Pour calculer le TID, la valeur ":ss" doit être tronquée de l'heure réelle.

Des exemples de valeurs calculées de TID sont donnés dans le Tableau 15.

Tableau 15 – Exemples de calcul de TID

Date d'émission:	Heure d'émission:	Minutes écoulées:	ID de jeton de 24 bits obtenu:
1 janvier 1993	00:00:00	0	0000 0000 0000 0000 0000 0000
1 janvier 1993	00:01:45	1	0000 0000 0000 0000 0000 0001
25 mars 1993	13:55:22	120 355	0000 0001 1101 0110 0010 0011
25 mars 1996	13:55:22	1 698 595	0001 1001 1110 1011 0010 0011
1 novembre 2005	00:01:55	6 749 281	0110 0110 1111 1100 0110 0001
1 décembre 2015	00:01:05	12 051 361	1011 0111 1110 0011 1010 0001
24 novembre 2024	20:15:00	16 777 215	1111 1111 1111 1111 1111 1111
1 janvier 2014	00:00:00	0	0000 0000 0000 0000 0000 0000
24 novembre 2045	20:15:00	16 777 215	1111 1111 1111 1111 1111 1111
1 janvier 2035	00:00:00	0	0000 0000 0000 0000 0000 0000
24 novembre 2066	20:15:00	16 777 215	1111 1111 1111 1111 1111 1111

Afin d'éviter la réutilisation d'un jeton lorsqu'un changement de date de référence est effectué, certaines procédures opérationnelles ont besoin d'être exécutées. Se référer à l'Article C.12 pour des informations complémentaires.

6.3.5.2 SpecialReservedTokenIdentifier

Le TokenIdentifier correspondant à 00 h 01 min de chaque jour est réservé pour des jetons d'application spéciaux et ne peut être utilisé pour aucun autre jeton.

En utilisant le format de date et heure AAAA:MM:JJ:hh:mm:ss, les valeurs de TID réservé correspondent à xxxx:xx:xx:00:01:xx.

Si un jeton, autre qu'un jeton d'application spécial, est à générer à une heure correspondant à ce TID réservé, alors 1 min doit être ajoutée au TID.

Voir aussi l'Article C.4 pour le Code de bonnes pratiques de gestion de ce TID réservé spécial.

NOTE L'utilisation de jetons d'application spéciaux est facultative (voir l'Article C.11), mais la règle relative à la façon d'utiliser le TID réservé spécial est obligatoire.

6.3.5.3 Plusieurs jetons générés dans la même minute

Le POS doit assurer qu'aucun jeton acheté légitimement ne peut porter le même TID que celui de n'importe quel autre jeton acheté légitimement pour le même compteur à paiement même si plus d'un jeton est acheté dans la même minute sur le même POS.

Si plusieurs jetons ont besoin d'être générés dans la même minute pour le même compteur à paiement, alors 1 min doit être ajoutée au TID de chaque jeton successif dans l'ensemble. À la fin du processus de génération de jeton, le POS doit retourner à nouveau à l'heure réelle.

Cela doit s'appliquer à tout jeton qui met en œuvre un TID.

Cela ne doit pas s'appliquer à des jetons d'application spéciaux qui mettent en œuvre le SpecialReservedTokenIdentifier (voir 6.3.5.2).

Par exemple: si 3 jetons de crédit A, B et C sont générés dans la même minute à 13h23 et dans l'ordre séquentiel A, B et C, alors A doit porter le marqueur temporel 13h23 du TID, B doit porter le marqueur temporel 13h24 et C doit porter 13h25.

6.3.6 Amount: TransferAmount

L'unité associée pour le montant du transfert est définie dans le Tableau 16.

Tableau 16 – Unités de mesure pour l'électricité

Type de transfert	Unités de mesure
Énergie électrique	Wattheures x 100 (0,1 kWh)
Puissance électrique	Watts

La STS Association réserve également les types de transfert donnés dans le Tableau 17 pour d'autres applications.

Tableau 17 – Unités de mesure pour d'autres applications

Type de transfert	Unités de mesure
Eau	Litres x 100
Gaz	Mètres cubes
Temps	Minutes
Monnaie	À l'étude
NOTE La STS Association définit d'autres types de transfert futurs pour d'autres services d'entreprise de distribution.	

Les 16 bits du champ TransferAmount (montant de transfert) sont subdivisés en deux sections, à savoir un exposant en base 10 de 2 bits et une mantisse de 14 bits. Les bits sont numérotés de droite à gauche, en commençant à 0. Le bit 15 est le bit de poids de fort de l'exposant et le bit 13 est le bit de poids fort de la mantisse. Les allocations de bits dans ce champ sont illustrées dans le Tableau 18.

Tableau 18 – Allocations des bits pour le TransferAmount

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

La formule pour la conversion des montants de transfert est comme suit:

$t = 10^{e} \times m$, pour $e = 0$; ou

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

où

t est le montant du transfert,

e est l'exposant en base 10,

m est la mantisse, et

n est un nombre entier dans la plage 1 à e inclus.

Toutes les conversions de montant de transfert doivent être arrondies en excès en faveur du consommateur. Les plages possibles des montants de transfert et les erreurs maximales associées qui peuvent se produire en raison de l'arrondi par excès sont montrées dans le Tableau 19. Des exemples de valeurs de TransferAmount sont donnés dans le Tableau 20.

Tableau 19 – Erreur maximale d'arrondi

Valeur de l'exposant	Plage du montant de transfert	Erreur maximale
0	0000000 à 00016383	0,000
1	0016384 à 00180214	0,061 %
2	0180224 à 01818524	0,055 %
3	1818624 à 18201624	0,055 %

Tableau 20 – Exemples de valeurs de TransferAmount pour le transfert de crédit

Élément	Unités achetées:	Champ montant de transfert de 16 bits obtenu	Unités converties et reçues par le compteur
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

6.3.7 CRC: CyclicRedundancyCode

Le CRC est un champ somme de contrôle utilisé pour vérifier l'intégrité des données transférées. La somme de contrôle est dérivée en utilisant le polynôme générateur de CRC suivant:

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

La longueur totale des données transférées par l'intermédiaire du jeton est de 66 bits. Les 16 derniers bits composent la somme de contrôle de CRC qui est dérivée des 50 bits qui les précèdent. Ces 50 bits sont complétés de 6 zéros binaires à gauche pour atteindre 56 bits. Avant calcul, la somme de contrôle de CRC est initialisée à FFFF hex. (Voir l'exemple dans le Tableau 21.)

Tableau 21 – Exemple de calcul de CRC

50 bits d'origine	0 00 4A 2D 90 0F F2 hex
Complétés à gauche pour obtenir 7 octets	00 00 4A 2D 90 0F F2 hex
Somme de contrôle calculée	0F FA hex

6.3.8 Control: InitiateMeterTest/DisplayControlField

Le champ Initier les données d'essai de compteur à paiement a une longueur de 36/28 bits et il est utilisé pour indiquer le type d'essai à réaliser. L'essai particulier est sélectionné en mettant à la valeur logique UN le bit approprié. Les valeurs admissibles du champ sont définies dans le Tableau 22.

Tableau 22 – Valeurs admissibles du champ Control

LS Bit No. = 1	Essai n°	Action	Condition
Tous les bits = 1	0	Effectuer l'essai n° 2 à n° 5 plus, facultativement, n'importe quel autre	Obligatoire
1	1	Soumettre à essai le commutateur de charges	Facultatif
2	2	Soumettre à essai les dispositifs d'affichage d'informations de compteur à paiement	Obligatoire
3	3	Afficher les totaux cumulés des registres d'énergie en kWh	Obligatoire
4	4	Afficher le KRN	Obligatoire
5	5	Afficher le TI	Obligatoire
6	6	Soumettre à essai le dispositif lecteur de jeton	Facultatif
7	7	Afficher la limite de puissance maximale	Facultatif
8	8	Afficher le statut de fraude	Facultatif
9	9	Afficher la consommation d'énergie	Facultatif
10	10	Afficher la version du logiciel	Facultatif
11	11	Afficher la limite de déséquilibre de puissance des phases	Facultatif
12	12	Afficher le facteur du compteur d'eau	Obligatoire pour le compteur à paiement d'eau
13	13	Afficher le taux de tarif	Obligatoire pour le compteur à paiement à monnaie
14-28/36	Réservé	Réservé par la STS Association pour une affectation future	Réservé
NOTE Le registre d'énergie cumulée en kWh est défini en 5.11.4 de l'IEC 62055-31:2005.			

Tous les compteurs à paiement doivent prendre en charge l'essai numéro 0; si l'un ou plusieurs des essais incorporés ne sont pas pris en charge, le compteur à paiement doit effectuer le sous-ensemble d'essais qui sont pris en charge.

Cette option est assujettie à l'abonnement passé entre le fournisseur et l'entreprise de distribution et ne doit pas constituer une partie normative de la présente Norme.

Si plus d'un essai est spécifié sur un seul et même jeton, le comportement du compteur à paiement doit faire l'objet d'un accord entre l'entreprise de distribution et le fournisseur et il ne doit pas constituer une partie normative de la présente Norme.

6.3.9 MPL: MaximumPowerLimit

Le champ "limite de puissance maximale" est un champ de 16 bits qui indique la puissance maximale qu'une charge peut tirer, en watts. Le calcul de ce champ est identique à celui du champ de TransferAmount (voir 6.3.6). Voir aussi la note en 8.6 pour les exigences fonctionnelles du MeterApplicationProcess.

6.3.10 MPPUL: MaximumPhasePowerUnbalanceLimit

Le champ "limite maximale de déséquilibre de puissance de phases" est un champ de 16 bits qui indique la différence maximale admissible de puissance entre les charges des phases, en watts Le calcul de ce champ est identique à celui du champ de TransferAmount (voir 6.3.6).

6.3.11 Rate: TariffRate

Réservé par la STS Association pour une définition future.

6.3.12 WMFactor: WaterMeterFactor

Réservé par la STS Association pour une application relative à l'eau.

6.3.13 Register: RegisterToClear

Valeur unique binaire de 16 bits dans la plage 0 à FFFF hex; pour sélectionner le registre particulier qu'il convient de vider avec le jeton ClearCredit. Les valeurs définies sont données dans le Tableau 23.

Tableau 23 – Sélection du registre à vider

Valeur	Action
0	Vider le registre Electricity Credit (crédit d'électricité)
1	Vider le registre Water Credit (crédit d'eau)
2	Vider le registre Gas Credit (crédit de gaz)
3	Vider le registre Time Credit (crédit de temps)
4	Vider le registre Currency Credit (crédit de monnaie)
5 à FFFE hex	Réservé pour une affectation future
FFFF hex	Vider tous les registres de crédit dans le compteur à paiement

6.3.14 NKHO: NewKeyHighOrder

Les 32 bits de poids fort de la nouvelle DecoderKey qui a été générée (voir 6.4.4) et qu'il faut transférer au compteur à paiement au moyen du jeton.

6.3.15 NKLO: NewKeyLowOrder

Les 32 bits de poids faible de la nouvelle DecoderKey qui a été générée (voir 6.4.5) et qu'il faut transférer au compteur à paiement au moyen du jeton.

6.3.16 KENHO: KeyExpiryNumberHighOrder

Il s'agit des 4 bits de poids fort du KEN (voir 6.1.10).

6.3.17 KENLO: KeyExpiryNumberLowOrder

Il s'agit des 4 bits de poids faible du KEN (voir 6.1.10).

6.3.18 RO: RolloverKeyChange

Si le bit RolloverKeyChange est mis = 1, le compteur à paiement doit effectuer un changement de clé avec passage au zéro. Cette opération est identique à un changement de clé normal, excepté que le stockage en mémoire du TID dans le compteur à paiement est rempli d'identificateurs de jeton ayant la valeur 0 (zéro).

6.4 Fonctions de TCDUGeneration

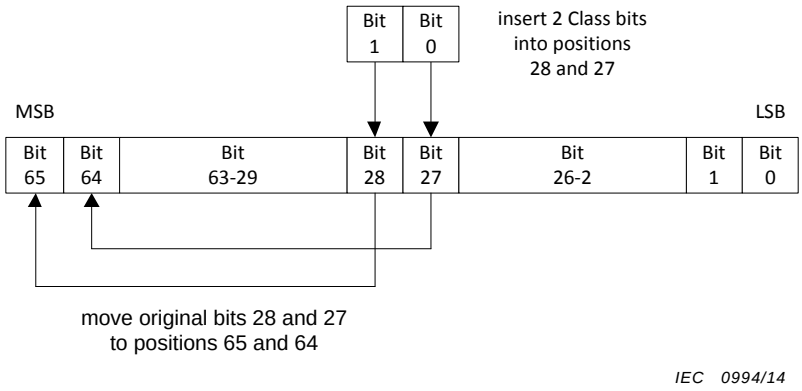
6.4.1 Définition de la TCDU

La TCDU peut être différente pour chaque TokenCarrierType et elle est donc définie séparément pour chaque norme de protocole de couche physique pertinente pour chaque partie de la série IEC 62055-5x.

6.4.2 Transposition des bits de Class (Classe)

Cette fonction est utilisée par d'autres fonctions de TCDUGeneration (voir 6.4.3 à 6.4.5). Elle insère les 2 bits de Class dans un train de données de 64 bits pour obtenir un nombre de 66 bits suivant la méthode présentée dans les grandes lignes ci-dessous.

Le nombre de 64 bits a son bit de poids faible placé à la position binaire 0 et son bit de poids fort placé à la position binaire 63. La chaîne du nombre binaire de 64 bits est modifiée pour inclure la Class de jetons non chiffrée. La valeur de la Class de jetons de 2 bits est insérée pour occuper les positions binaires 28 et 27. Les valeurs d'origine des positions binaires 28 et 27 sont déplacées aux positions binaires 65 et 64. Le bit de poids fort de la Class de jetons occupe maintenant la position binaire 28. Le processus est montré à la Figure 6.



Légende

Anglais	Français
insert 2 Class bits into positions 28 and 27	insérer aux positions 28 et 27 les 2 bits de Class
MSB	Bit de poids fort
LSB	Bit de poids faible
move original bits 28 and 27 to positions 65 and 64	déplacer vers les positions 65 et 64 les bits 28 et 27 d'origine

Figure 6 – Transposition des 2 bits de Class

Exemple: Insertion de la Class de jetons = 01 (binaire).

Le nombre binaire de 64 bits groupé en quartets: (Les bits 27 et 28 sont signalés en gras):

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

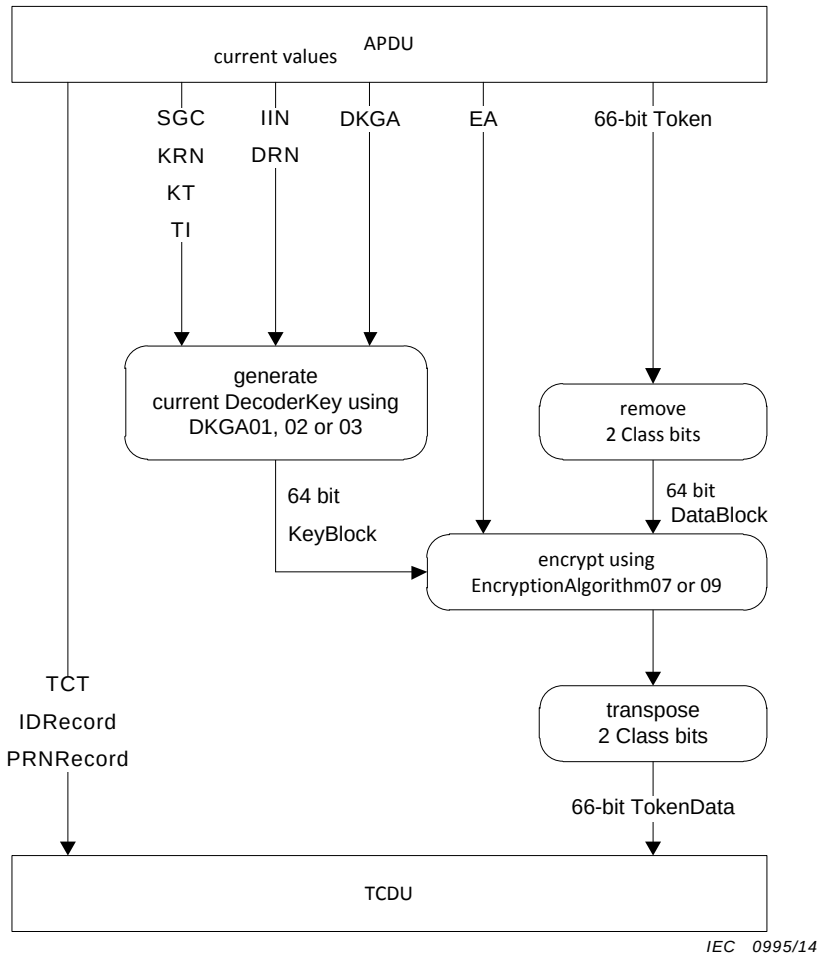
Copier les bits 28 et 27 aux positions binaires 65 et 64, créant ainsi un nombre de 66 bits:

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

Remplacer les bits 28 et 27 par les 2 bits de Class:

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

6.4.3 Fonction TCDUGeneration pour les jetons de Class 0,1 et 2



Légende

Anglais	Français
DataBlock	Bloc de données
TCDU	TCDU
APDU current values	APDU valeurs actuelles
encrypt using EncryptionAlgorithm07 or 09	Chiffrer en utilisant l'algorithme EncryptionAlgorithm07 ou 09
transpose 2 Class bits	Transposer les 2 bits de Class
64 bit KeyBlock	Bloc de clés de 64 bits
generate current DecoderKey using DKGA01, 02 or 03	Générer la clé de décodeur (DecoderKey) courante en utilisant l'Algorithme de génération de clé de décodeur DKGA01, DKGA 02 ou DKGA 03
Remove 2 Class bits	Retirer les 2 bits de Class
64 bit DataBlock	Bloc de données de 64 bits
EA	EA
DKGA	DKGA
SGC	SGC
KRN	KRN
KT	KT
TI	TI
IIN	IIN

Anglais	Français
DRN	DRN
66-bit Token	Jeton de 64 bits
TCT	TCT
IDRecord	Données d'enregistrement d'identification
PRNRecord	Données d'enregistrements pour impression
66-bit TokenData	Données du jeton de 66 bits

Figure 7 – Fonction TCDUGeneration pour les jetons de Class 0, 1 et 2

Il s'agit de la fonction de transfert de l'APDU vers la TCDU (voir Figure 7) et elle s'applique à tous les jetons de Class 0, Class 1 et Class 2, excepté les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey (voir 6.4.4 et 6.4.5).

NOTE 1 Les éléments de données dans l'APDU sont définis en 6.1.1.

NOTE 2 Les éléments de données dans la TCDU sont définis dans une partie de la norme de protocole de couche physique de la série IEC 62055-5x applicable au TCT spécifique d'intérêt.

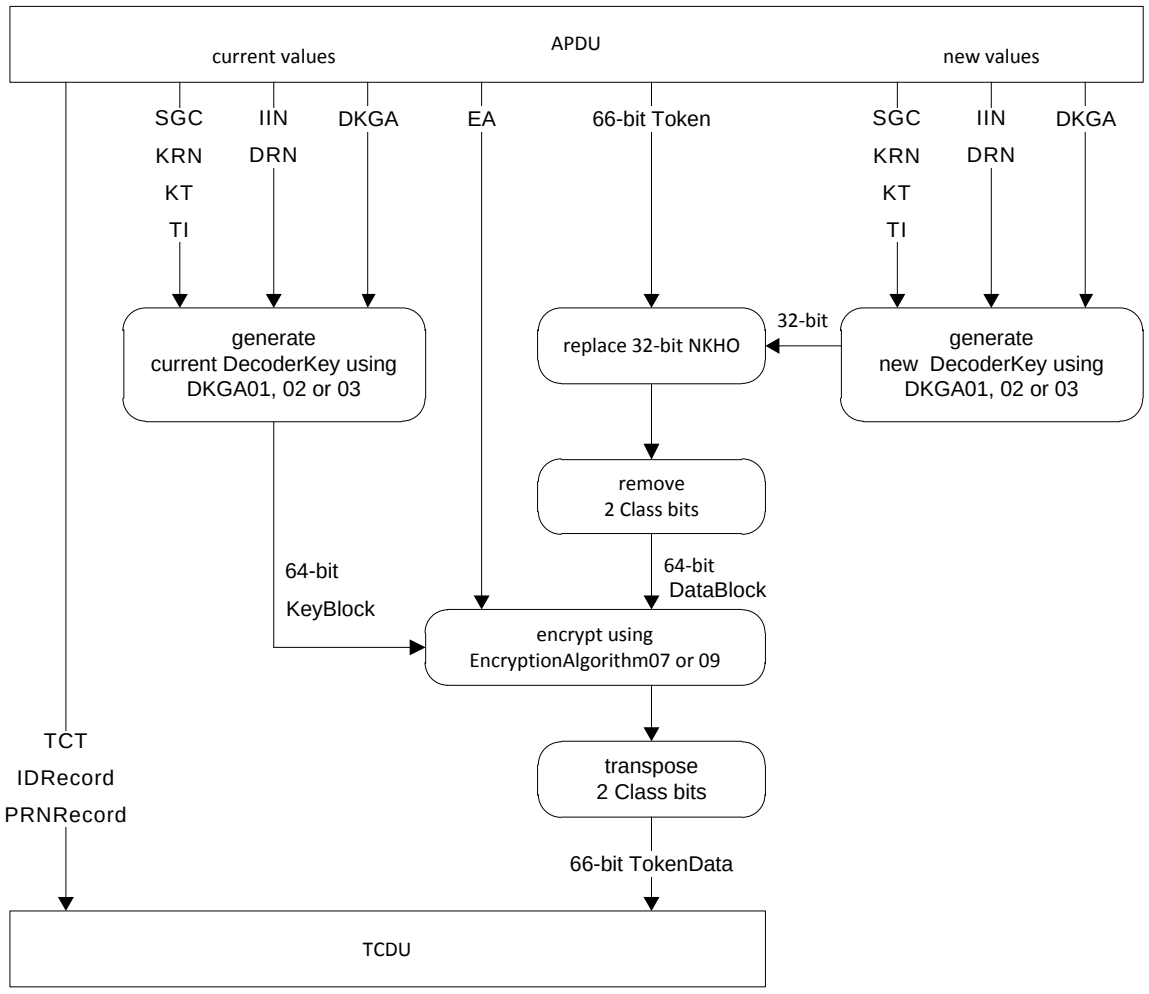
La fonction de transfert pour les jetons de Class 0 et Class 2 est présentée dans les grandes lignes comme suit:

- Les 2 bits de Class sont enlevés du jeton de 66 bits pour donner un résultat de 64 bits, qui est alors présenté à l'algorithme de chiffrement comme étant sa donnée d'entrée de DataBlock. L'algorithme spécifique à utiliser est fonction du code EA dans l'APDU;
- L'entrée KeyBlock pour l'algorithme de chiffrement est obtenue à partir de l'algorithme de génération de clé de décodeur, qui génère la DecoderKey courante en utilisant les valeurs courantes de SGC, KRN, KT, TI, IIN et DRN issues de l'APDU comme indiqué. L'algorithme spécifique de génération de clé de décodeur à utiliser est fonction de la valeur de DKGA dans l'APDU;
- Après chiffrement, les 2 bits de Class sont réinsérés dans le nombre de 64 bits selon la méthode définie en 6.4.2 pour donner un résultat de 66 bits, qui est peuplé dans le champ TokenData de la TCDU selon la définition particulière donnée dans la norme de protocole de couche physique pertinente;
- De même, les éléments de données TCT, IDRecord et PRNRecord issus de l'APDU sont transférés vers la TCDU comme indiqué, dans les champs appropriés de la TCDU en fonction de la définition particulière donnée dans la norme de protocole de couche physique pertinente;

La fonction de transfert pour les jetons de Class 1 est identique à la fonction de TCDUGeneration pour les jetons de Class 0 et de Class 2, excepté que le jeton n'est pas chiffré. Cette fonction est présentée dans les grandes lignes comme suit:

- Les 2 bits de Class sont retirés du jeton de 66 bits et transposés selon la méthode définie en 6.4.2 pour donner un résultat de 66 bits, qui est peuplé dans le champ TokenData de la TCDU selon la définition particulière donnée dans la norme de protocole de couche physique pertinente;
- De même, les éléments de données TCT, IDRecord et PRNRecord issus de l'APDU sont transférés vers la TCDU comme indiqué, dans les champs appropriés de la TCDU en fonction de la définition particulière donnée dans la norme de protocole de couche physique pertinente.

6.4.4 Fonction TCDUGeneration pour le jeton Set1stSectionDecoderKey



IEC 0996/14

Légende

Anglais	Français
APDU current values	APDU valeurs actuelles
new values	nouvelles valeurs
66-bit Token	Jeton de 66 bits
generate current DecoderKey using DKGA01, 02 or 03	générer la clé de décodeur (DecoderKey) courante en utilisant l'Algorithme de génération de clé de décodeur DKGA01, DKGA 02 ou DKGA 03
replace 32-bit NKHO	Remplacer le NKHO (bits de poids fort de nouvelle clé) de 32 bits
32-bit	32 bits
generate new DecoderKey using DKGA01, 02 or 03	générer la nouvelle clé de décodeur (DecoderKey) en utilisant l'Algorithme de génération de clé de décodeur DKGA01, DKGA 02 ou DKGA 03
remove 2 Class bits	retirer les 2 bits de Class
64-bit KeyBlock	Bloc de clés de 64 bits
64-bit DataBlock	Bloc de données de 64 bits
encrypt using EncryptionAlgorithm07 or 09	Chiffrer en utilisant l'algorithme de chiffrement EncryptionAlgorithm07 ou 09
transpose 2 Class bits	transposer les 2 bits de Class
66-bit TokenData	Données du jeton de 66 bits

Anglais	Français
EA	EA
DKGA	DKGA
SGC	SGC
KRN	KRN
KT	KT
TI	TI
IIN	IIN
DRN	DRN
TCT	TCT
IDRecord	Données d'enregistrement d'identification
PRNRecord	Données d'enregistrements pour impression
TCDU	TCDU

Figure 8 – Fonction TCDUGeneration pour le jeton Set1stSectionDecoderKey

Il s'agit de la fonction de transfert de l'APDU vers la TCDU (voir Figure 8) et elle est applicable seulement au jeton Set1stSectionDecoderKey.

La fonction TCDUGeneration Set1stSectionDecoderKey est montrée comme étant distincte de la fonction TCDUGeneration Set2ndSectionDecoderKey, mais, dans la pratique, les deux peuvent être fusionnées en une seule afin d'économiser des ressources de traitement et par souci de commodité. Dans un tel cas, la génération de la nouvelle DecoderKey a seulement besoin de se produire une fois par exemple, bien que le résultat final soit toujours le même. Ainsi, deux instances de TCDU distinctes sont toujours produites: une pour le jeton Set1stSectionDecoderKey et une deuxième pour le jeton Set2ndSectionDecoderKey.

Noter qu'il faut que l'APDU propose deux jeux de données pour le PANBlock et le CONTROLBlock: un jeu avec les nouvelles données pour la nouvelle DecoderKey et un deuxième jeu avec les données courantes pour la DecoderKey courante. La valeur de DKGA est la même pour les deux jeux.

NOTE 1 Les éléments de données dans l'APDU sont définis en 6.1.1.

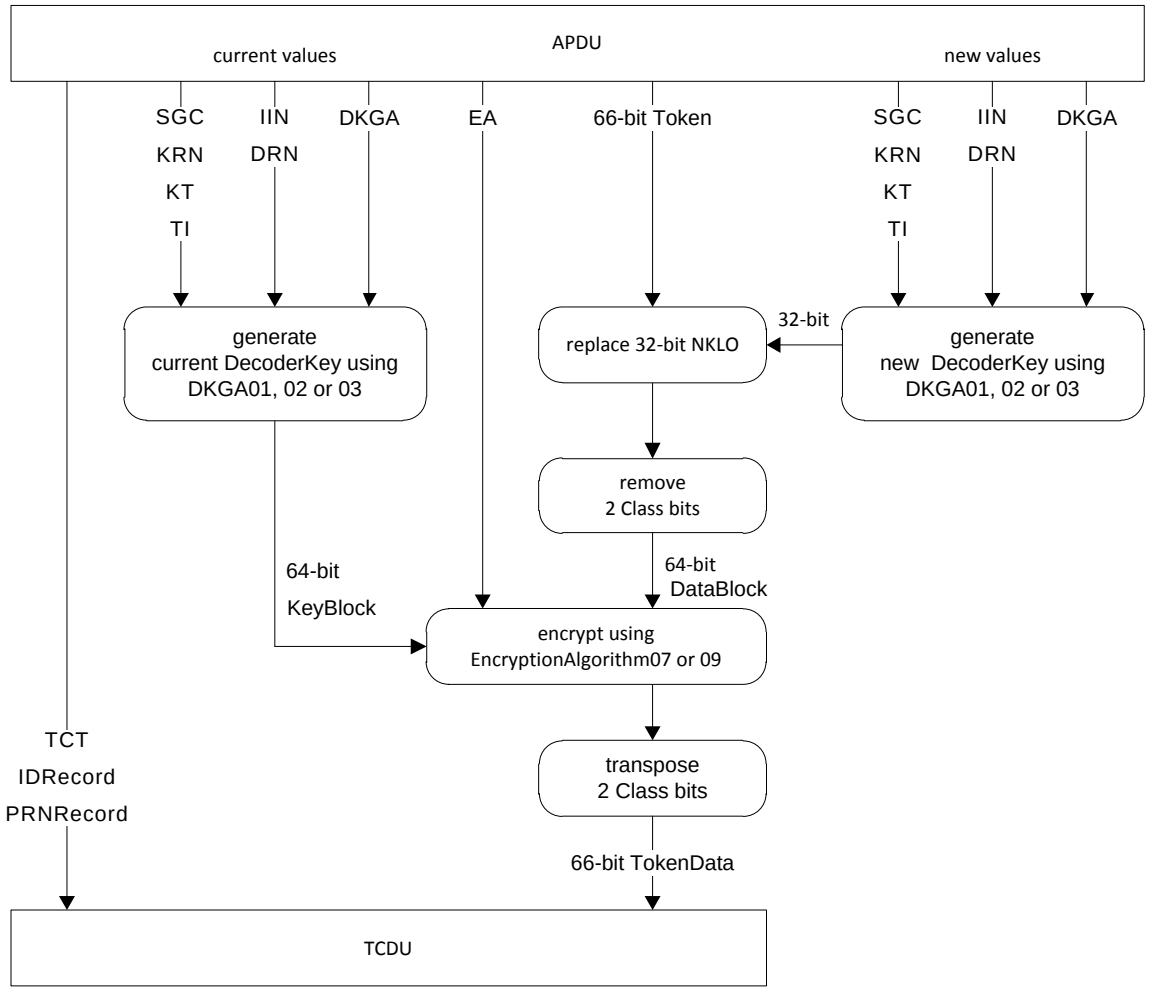
NOTE 2 Les éléments de données dans la TCDU sont définis dans chaque partie de la norme de protocole de couche physique de la série IEC 62055-5x applicable au TCT spécifique d'intérêt.

Cette fonction de transfert est présentée dans les grandes lignes comme suit:

- La nouvelle DecoderKey est générée à l'aide des nouvelles valeurs de SGC, KRN, KT, TI, IIN et DRN. L'algorithme spécifique à utiliser est fonction de la valeur de DKGA dans l'APDU.
- Les 32 bits de poids fort de la valeur de la nouvelle DecoderKey obtenue sont ensuite utilisés pour remplacer le champ NKHO du jeton Set1stSectionDecoderKey (voir 6.2.7) tel que présenté par l'APDU;
- Les 2 bits de Class sont enlevés du jeton de 66 bits pour donner un résultat de 64 bits, qui est alors présenté à l'algorithme de chiffrement comme étant sa donnée d'entrée de DataBlock. L'algorithme de chiffrement spécifique à utiliser est fonction du code EA dans l'APDU;
- l'entrée KeyBlock pour l'algorithme de chiffrement est obtenue à partir de l'algorithme de génération de clé de décodeur, qui génère la DecoderKey courante en utilisant les valeurs courantes de SGC, KRN, KT, TI, IIN et DRN issues de l'APDU comme indiqué. L'algorithme spécifique de génération de clé de décodeur à utiliser est fonction de la valeur de DKGA dans l'APDU;

- après chiffrement, les 2 bits de Class sont réinsérés dans le nombre de 64 bits selon la méthode définie en 6.4.2 pour donner un résultat de 66 bits, qui est peuplé dans le champ TokenData de la TCDU selon la définition particulière donnée dans la norme de protocole de couche physique pertinente;
- de même, les éléments de données TCT, IDRecord et PRNRecord issus de l'APDU sont transférés vers la TCDU comme indiqué, dans les champs appropriés de la TCDU en fonction de la définition particulière donnée dans la norme de protocole de couche physique pertinente.

6.4.5 Fonction TCDUGeneration pour le jeton Set2ndSectionDecoderKey



IEC 0997/14

Légende

Anglais	Français
APDU current values new values	APDU valeurs actuelles nouvelles valeurs
66-bit Token	Jeton de 66 bits
generate current DecoderKey using DKGA01, 02 or 03	générer la clé de décodeur (DecoderKey) courante en utilisant l'Algorithme de génération de clé de décodeur DKGA01, DKGA 02 ou DKGA 03
replace 32-bit NKLO	Remplacer le NKLO (bits de poids faible de nouvelle clé) de 32 bits
32-bit	32 bits
generate new DecoderKey using DKGA01, 02 or 03	générer la nouvelle clé de décodeur (DecoderKey) en utilisant l'Algorithme de génération de clé de décodeur DKGA01, DKGA 02 ou DKGA 03

remove 2 Class bits	retirer les 2 bits de Class
64-bit KeyBlock	Bloc de clés de 64 bits
64-bit DataBlock	Bloc de données de 64 bits
encrypt using EncryptionAlgorithm07 or 09	Chiffrer en utilisant l'algorithme de chiffrement EncryptionAlgorithm07 ou 09
transpose 2 Class bits	transposer les 2 bits de classe
66-bit TokenData	Données du jeton de 66 bits
EA	EA
DKGA	DKGA
SGC	SGC
KRN	KRN
KT	KT
TI	TI
IIN	IIN
DRN	DRN
TCT	TCT
IDRecord	Données d'enregistrement d'identification
PRNRecord	Données d'enregistrements pour impression
TCDU	TCDU

Figure 9 – Fonction TCDUGeneration pour le jeton Set2ndSectionDecoderKey

Il s'agit de la fonction de transfert de l'APDU vers la TCDU (voir Figure 9) et elle est applicable seulement au jeton Set2ndSectionDecoderKey.

La fonction TCDUGeneration Set2ndSectionDecoderKey est montrée comme étant distincte de la fonction TCDUGeneration Set1stSectionDecoderKey, mais, dans la pratique, les deux peuvent être fusionnées en une seule afin d'économiser des ressources de traitement et par souci de commodité. Dans un tel cas, la génération de la nouvelle DecoderKey a seulement besoin de se produire une fois par exemple, bien que le résultat final soit toujours le même. Ainsi, deux instances de TCDU distinctes sont toujours produites: une pour le jeton Set1stSectionDecoderKey et une deuxième pour le jeton Set2ndSectionDecoderKey.

Noter qu'il faut que l'APDU propose deux jeux de données pour le PANBlock et le CONTROLBlock: un jeu avec les nouvelles données pour la nouvelle DecoderKey et un deuxième jeu avec les données courantes pour la DecoderKey courante. La valeur de DKGA est la même pour les deux jeux.

NOTE 1 Les éléments de données dans l'APDU sont définis en 6.1.1.

NOTE 2 Les éléments de données dans la TCDU sont définis dans chaque partie de la norme de protocole de couche physique de la série IEC 62055-5x applicable au TCT spécifique d'intérêt.

Cette fonction de transfert est présentée dans les grandes lignes comme suit:

- La nouvelle DecoderKey est générée à l'aide des nouvelles valeurs de SGC, KRN, KT, TI, IIN et DRN. L'algorithme spécifique de génération de clé de décodeur à utiliser est fonction de la valeur de DKGA dans l'APDU;
- Les 32 bits de poids faible de la valeur de la nouvelle DecoderKey obtenue sont ensuite utilisés pour remplacer le champ NKLO du jeton Set2ndSectionDecoderKey (voir 6.2.8) tel que présenté par l'APDU;
- Les 2 bits de Class sont enlevés du jeton de 66 bits pour donner un résultat de 64 bits, qui est alors présenté à l'algorithme de chiffrement comme étant sa donnée d'entrée de

DataBlock. L'algorithme de chiffrement spécifique à utiliser est fonction du code EA dans l'APDU;

- l'entrée KeyBlock pour l'algorithme de chiffrement est obtenue à partir de l'algorithme de génération de clé de décodeur, qui génère la DecoderKey courante en utilisant les valeurs courantes de SGC, KRN, KT, TI, IIN et DRN issues de l'APDU comme indiqué. L'algorithme spécifique de génération de clé de décodeur à utiliser est fonction de la valeur de DKGA dans l'APDU;
- après chiffrement, les 2 bits de Class sont réinsérés dans le nombre de 64 bits selon la méthode définie en 6.4.2 pour donner un résultat de 66 bits, qui est peuplé dans le champ TokenData de la TCDU selon la définition particulière donnée dans la norme de protocole de couche physique pertinente;
- de même, les éléments de données TCT, IDRecord et PRNRecord issus de l'APDU sont transférés vers la TCDU comme indiqué, dans les champs appropriés de la TCDU en fonction de la définition particulière donnée dans la norme de protocole de couche physique pertinente.

6.5 Fonctions de sécurité

6.5.1 Exigences générales

À l'exception des valeurs de DITK, les valeurs de VendingKey et de DecoderKey doivent seulement être générées par un dispositif chargé de la génération de jetons, tel qu'un POS qui est certifié comme étant conforme à la STS et qui est assujetti à un KeyManagementSystem certifié STS (voir Article 9). Le présent paragraphe décrit les méthodes de génération de clé utilisées par de tels dispositifs et il est applicable aux constructeurs de ces dispositifs.

6.5.2 Attributs de clé et changements de clé

6.5.2.1 Exigences relatives au changement de clé

À l'exception des valeurs de DITK, les valeurs des clés STS doivent seulement être introduites ou modifiées dans un compteur à paiement à partir du dispositif chargé de la gestion de clé, tel qu'un POS qui est certifié comme étant conforme à la STS et qui est assujetti à la gestion de clé STS. Le présent paragraphe décrit la méthode de changement de clé de STS utilisée entre de tels dispositifs et les compteurs à paiement, et il est applicable aux constructeurs de ces dispositifs et compteurs à paiement.

Un changement de clé de STS fournit le mécanisme pour changer la DecoderKey présente dans un décodeur en la faisant passer de sa valeur courante à sa nouvelle valeur. Ce processus peut être initié par plusieurs événements ou circonstances, y compris ce qui suit:

- un compteur à paiement, neuf ou réparé, qui contient la valeur de DITK d'un constructeur doit être changé avant de quitter les locaux de fabrication ou de réparation pour contenir la valeur appropriée par défaut (DDTK) du constructeur ou de la DecoderKey (DUTK ou DCTK) de l'entreprise de distribution selon le SupplyGroup auquel le compteur à paiement a été alloué;
- une VendingKey d'un SupplyGroup a expiré ou a été compromise et elle est remplacée par une nouvelle révision de VendingKey et, donc, chaque DecoderKey au sein du SupplyGroup doit être changée en faisant passer sa valeur de DecoderKey courante à la valeur de DecoderKey qui correspond à la valeur de la nouvelle VendingKey;
- un compteur à paiement est réalloué d'un SupplyGroup à un autre SupplyGroup et, donc, sa DecoderKey doit être changée en faisant passer sa valeur courante générée à partir de la VendingKey du SupplyGroup précédent à la nouvelle valeur générée à partir de la VendingKey de son nouveau SupplyGroup; ou
- le TI pour un compteur à paiement a changé et, donc, sa DecoderKey doit être changée en faisant passer sa valeur courante (qui correspond au TI précédent) à la nouvelle valeur (qui correspond au nouveau TI).

La paire de jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey opère un changement de clé de STS. Cette paire de jetons de gestion spécifique à un compteur transfère les informations suivantes, du POS vers le compteur à paiement, chiffrées sous la DecoderKey courante:

- la valeur de la nouvelle DecoderKey;
- le KEN;
- le KRN;
- le KT;
- le TI.

Un changement de clé STS pour un compteur à paiement doit être initié automatiquement chaque fois que l'un quelconque des attributs suivants de la VendingKey change de valeur:

- la valeur de la VendingKey;
- la valeur du SGC;
- la valeur du TI;
- la valeur du KEN;
- la valeur du KRN;
- La valeur du KT.

NOTE Voir 6.1.1 pour les spécifications particulières relatives aux éléments de données dans l'APDU.

6.5.2.2 Classification des VendingKey

6.5.2.2.1 Classification des VendingKey (clés de vente)

La VendingKey est une valeur clé DES qui est secrètement générée, stockée et distribuée dans le KeyManagementSystem (voir Annexe A). Les VendingKey DES sont les clés-germes à partir desquelles les DecoderKey sont générées.

La VendingKey est classée selon la valeur de son KT associé, qui est un attribut qui définit le but dans lequel la clé peut être utilisée. Trois valeurs de KT sont définies pour les VendingKey et correspondent à trois des types de SupplyGroup (voir 6.1.6), à savoir Default (c'est-à-dire: valeur par défaut), Unique et Common (c'est-à-dire: commune). La VendingKey pour un SupplyGroup donné est la clé-germe utilisée pour générer des valeurs de DecoderKey pour tous les compteurs à paiement au sein du SupplyGroup.

Les VendingKey STS sont classées selon les valeurs de KT données dans le Tableau 24.

Tableau 24 – Classification des VendingKey (clés de vente)

KT	Type de SGC	Type de VendingKey	Contexte
0	Initialisation	Non spécifié	Sans objet
1	Default (valeur par défaut)	VDDK	VendingDefaultDESKey (Clé DES par défaut de vente)
2	Unique	VUDK	VendingUniqueDESKey (Clé DES unique de vente)
3	Common (commun)	VCDK	VendingCommonDESKey (Clé DES commune de vente)

À un instant donné quelconque, une valeur unique de VDDK existe pour chaque SupplyGroup de type "Default" défini. De même, une valeur unique de VUDK pour chaque SupplyGroup de type "Unique" et une valeur unique de VCDK pour chaque SupplyGroup de type "Common" sont définies.

6.5.2.2.2 VDDK: VendingDefaultDESKey

Ce type de clé est utilisé comme clé-germe pour générer des valeurs de DDTK – il ne doit pas être utilisé pour générer des valeurs de DITK, de DUTK ou de DCTK.

6.5.2.2.3 VUDK: VendingUniqueDESKey

Ce type de clé est utilisé comme clé-germe pour générer des valeurs de DUTK – il ne doit pas être utilisé pour générer des valeurs de DITK, de DDTK ou de DCTK.

6.5.2.2.4 VCDK: VendingCommonDESKey

Ce type de clé est utilisé comme clé-germe pour générer des valeurs de DCTK – il ne doit pas être utilisé pour générer des valeurs de DITK, de DDTK ou de DUTK.

6.5.2.3 Classification des DecoderKey

6.5.2.3.1 Classification des DecoderKey (clés de décodeur)

Les DecoderKey STS sont classées selon les valeurs de KT données dans le Tableau 25 et héritent de leur type de celui de la VendingKey, à partir de laquelle elles sont dérivées.

Tableau 25 – Classification des DecoderKey (clés de décodeur)

KT	Type de SGC	Type de DecoderKey	Contexte
0	Initialisation	DITK	DecoderInitialisationTransferKey
1	Default (valeur par défaut)	DDTK	DecoderDefaultTransferKey (Clé de transfert par défaut de décodeur)
2	Unique	DUTK	DecoderUniqueTransferKey (Clé de transfert unique de décodeur)
3	Common (commun)	DCTK	DecoderCommonTransferKey (clé de transfert commune de décodeur)

Pour de plus amples informations concernant les règles pour changer d'une clé d'un type à un autre type, voir la Figure 10 et le Tableau 26 en 6.5.2.4.

Un compteur à paiement doit être capable de stocker au moins une valeur de DecoderKey et la valeur de son KT associé dans son DecoderKeyRegister (voir 7.3.2).

Il ne doit pas être possible de lire ou d'extraire la valeur de DecoderKey à partir d'un compteur à paiement en toute circonstance, qu'elle soit chiffrée ou en texte clair.

6.5.2.3.2 DITK: DecoderInitialisationTransferKey

Les valeurs de DITK sont utilisées pour initialiser le DecoderKeyRegister pendant la production ou la réparation dans les locaux du constructeur. Ces clés sont la propriété du MeterManufacturer. À ce titre, elles sont générées et gérées par le constructeur, et sont inconnues de l'entreprise de distribution.

Aucun compteur à paiement acheté par l'entreprise de distribution ne doit quitter les locaux d'un constructeur avec une valeur de DITK dans le DecoderKeyRegister. Le DecoderKeyRegister doit contenir une valeur de DDTK, de DUTK ou de DCTK fournie par le KMC. Une DITK est le seul type de clé qui peut être introduite dans un compteur à paiement sous la forme d'une valeur en texte clair. Les valeurs de DDTK, de DUTK ou de DCTK ne peuvent être introduites dans un compteur à paiement que sous la forme de valeurs (chiffrées) de texte de chiffrement.

Une DITK ne doit être utilisée que pour les fonctions de gestion de clé ci-après:

- comme la clé parente d'une autre DITK; autrement dit, pour chiffrer une autre DITK dans le but de l'introduire dans le DecoderKeyRegister;
- comme la clé parente d'une DDTK;
- comme la clé parente d'une DUTK, et
- comme la clé parente d'une DCTK, mais seulement dans un compteur à paiement utilisant une carte magnétique effaçable comme support de jeton (pour la valeur de TCT = 01).

Les fonctions ci-dessus peuvent être accomplies par l'intermédiaire des jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey ou par l'intermédiaire d'un mécanisme de chargement propriétaire du constructeur qui utilise les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey. Il convient que le compteur à paiement n'accepte que la DDTK, DUTK ou DCTK chiffrée sous la DITK fournie par le constructeur au format de jeton Set1stSectionDecoderKey et Set2ndSectionDecoderKey.

Il est de la responsabilité du constructeur d'assurer que des mesures de sécurité appropriées sont appliquées à toute DITK afin que les valeurs de DDTK, de DUTK ou de DCTK chiffrées avec une DITK ne puissent pas être compromises.

Une DITK peut également être utilisée pour déchiffrer d'autres fonctions de gestion spécifiques à un compteur. Elle peut être utilisée pour déchiffrer une fonction de transfert de crédit STS; autrement dit, un jeton de TransferCredit STS valide peut être déchiffré et appliqué par un compteur à paiement qui contient une DITK dans son registre de clés afin de faciliter les essais du compteur à paiement pendant la production ou la réparation.

6.5.2.3.3 DDTK: DecoderDefaultTransferKey

Les valeurs de DDTK sont utilisées pour prendre en charge des compteurs à paiement alloués à un SupplyGroup par défaut. Un compteur à paiement qui n'a pas été alloué à un SupplyGroup de type Common ou à un SupplyGroup de type Unique au moment de la fabrication ou de la réparation ne peut pas être chargé avec sa valeur correspondante de DCTK ou de DUTK. À la place, il est alloué à un groupe par défaut (Default) propre à chaque constructeur et chargé avec sa valeur de DDTK correspondante. Chaque MeterManufacturer reçoit une VDDK unique, à partir de laquelle il génère toutes les valeurs de DDTK pour l'installation dans des compteurs à paiement pendant la fabrication.

Ultérieurement, au moment de l'installation ou de l'exploitation, un compteur à paiement qui a été maintenant réalloué à un autre SupplyGroup spécifique peut être chargé avec la valeur de DUTK ou de DCTK correspondante, chiffrée sous sa DDTK parente. Les valeurs de DDTK sont la propriété du MeterManufacturer respectif et sont gérées au sein du KeyManagementSystem.

Une DDTK est une valeur secrète, et ne doit pas être acceptée par un compteur à paiement sous la forme d'une valeur en texte clair. Un compteur à paiement ne doit charger une DDTK que si elle est chiffrée avec la DecoderKey parente présente dans le DecoderKeyRegister.

Une DDTK ne doit être utilisée que pour les fonctions de gestion de clé ci-après:

- comme la clé parente d'une autre DDTK; autrement dit, pour chiffrer une autre DDTK dans le but de l'introduire dans le DecoderKeyRegister;
- comme la clé parente d'une DUTK, et
- comme la clé parente d'une DCTK, mais seulement dans un compteur à paiement utilisant une carte magnétique effaçable comme support de jeton (pour la valeur de TCT = 01).

Les fonctions ci-dessus peuvent être accomplies par l'intermédiaire des jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey ou par l'intermédiaire d'un mécanisme de chargement propriétaire du constructeur qui utilise les

jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey. Une DDTK ne doit pas être utilisée pour déchiffrer une DITK dans le but de l'introduire dans le DecoderKeyRegister.

Une DDTK peut également être utilisée pour déchiffrer d'autres fonctions de gestion spécifiques à un compteur. Elle ne doit pas être utilisée pour déchiffrer et accepter une fonction de transfert de crédit STS; autrement dit, un jeton de TransferCredit valide ne doit pas être accepté par un compteur à paiement qui contient une DDTK dans son DKR, même si le jeton de TransferCredit a été chiffré avec la même valeur de DDTK.

NOTE L'accent est mis sur l'acceptation et non sur le déchiffrement du jeton de TransferCredit.

De même, un dispositif POS utilisé pour déchiffrer des jetons ne doit pas chiffrer les jetons de TransferCredit en utilisant des valeurs de DDTK (voir aussi 6.5.2.4).

6.5.2.3.4 DUTK: DecoderUniqueTransferKey

Les valeurs de DUTK sont utilisées pour prendre en charge des compteurs à paiement alloués à un SupplyGroup unique. Un compteur à paiement qui a été alloué à un SupplyGroup unique au moment de la fabrication ou de la réparation peut être chargé avec sa valeur de DUTK qui correspond au groupe unique et qui a été chiffrée sous une DITK parente. Ultérieurement, au moment de l'installation ou de l'exploitation, un compteur à paiement à réallouer à un autre groupe unique peut être chargé avec la valeur de DUTK correspondante, chiffrée sous une DUTK parente.

Une DUTK est une valeur secrète, et ne doit pas être acceptée par un compteur à paiement sous la forme d'une valeur en texte clair. Un compteur à paiement ne doit charger une DUTK que si elle a été chiffrée avec la DecoderKey parente présente dans le DecoderKeyRegister. Les valeurs de DUTK sont la propriété de l'entreprise de distribution respective et sont gérées au sein du KeyManagementSystem.

Un compteur à paiement acheté ou réparé qui quitte les locaux du constructeur peut contenir une valeur de DUTK fournie par le KMC dans le DecoderKeyRegister.

Une DUTK ne doit être utilisée que pour les fonctions de gestion de clé ci-après:

- comme la clé parente d'une autre DUTK; autrement dit, pour chiffrer une autre DUTK dans le but de l'introduire dans le DecoderKeyRegister, et
- comme la clé parente d'une DDTK.

Les fonctions ci-dessus peuvent être accomplies par l'intermédiaire des jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey ou par l'intermédiaire d'un mécanisme de chargement propriétaire du constructeur qui utilise les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey. Une DUTK ne doit pas être utilisée pour déchiffrer une DITK ou une DCTK dans le but de la charger dans le DecoderKeyRegister. De même, une DUTK ne doit pas être utilisée pour chiffrer une DITK ou une DCTK dans le but de la transférer vers le compteur à paiement sous la forme d'un jeton.

Une DUTK peut également être utilisée pour chiffrer ou déchiffrer d'autres fonctions de gestion spécifiques à un compteur. Elle peut être utilisée pour chiffrer ou déchiffrer une fonction de transfert de crédit STS; autrement dit, un jeton de TransferCredit valide peut être chiffré ou déchiffré et appliqué par un compteur à paiement qui contient une DUTK dans son DKR.

6.5.2.3.5 DCTK: DecoderCommonTransferKey

Les valeurs de DCTK sont utilisées pour prendre en charge les compteurs à paiement qui utilisent des supports de jeton de cartes magnétiques effaçables (c'est-à-dire: la valeur de TCT = 01) et qui sont alloués à des SupplyGroup communs. Un compteur à paiement qui a été alloué à un SupplyGroup commun au moment de la fabrication ou de la réparation peut

être chargé avec la valeur de DCTK qui correspond au SupplyGroup commun et qui a été chiffrée sous une DITK parente. Ultérieurement, au moment de l'installation ou de l'exploitation, un compteur à paiement à réallouer à un autre SupplyGroup commun peut être chargé avec la valeur de DCTK correspondante qui a été chiffrée sous une DCTK parente.

Une DCTK ne doit être utilisée qu'avec les compteurs à paiement qui utilisent les supports de jeton de cartes magnétiques effaçables (valeur de TCT = 01) et elle ne doit être acceptée que par de tels compteurs à paiement. Les compteurs à paiement avec tout autre type de support de jeton (valeur de TCT > 01) doivent rejeter les jetons chiffrés sous des valeurs de DCTK.

Les dispositifs de chiffrement POS ne doivent pas chiffrer les jetons utilisant des valeurs de DCTK autres que pour les supports de jetons de cartes magnétiques effaçables (valeur de TCT = 01).

Une DCTK est une valeur secrète, et ne doit pas être acceptée par un compteur à paiement sous la forme d'une valeur en texte clair. Un compteur à paiement ne doit charger une DCTK que si elle a été chiffrée avec la DecoderKey parente présente dans le DecoderKeyRegister. Les valeurs de DCTK sont la propriété de l'entreprise de distribution respective et sont gérées au sein du KeyManagementSystem.

Un compteur à paiement acheté ou réparé avec un support de jeton de carte magnétique effaçable (valeur de TCT = 01) qui quitte les locaux du constructeur peut contenir une valeur de DCTK fournie par le KMC dans le DecoderKeyRegister.

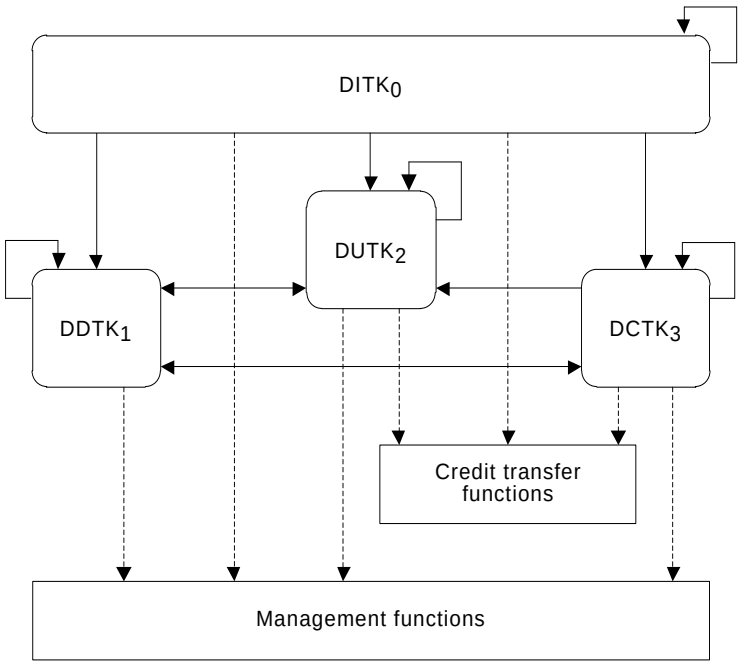
Une DCTK ne doit être utilisée que pour les fonctions de gestion de clé ci-après:

- comme la clé parente d'une autre DCTK; autrement dit, pour chiffrer une autre DCTK dans le but de l'introduire dans le DecoderKeyRegister;
- comme la clé parente d'une DDTK, et
- comme la clé parente d'une DUTK.

Les fonctions ci-dessus peuvent être accomplies par l'intermédiaire des jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey ou par l'intermédiaire d'un mécanisme de chargement propriétaire du constructeur qui utilise les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey. Une DCTK ne doit pas être utilisée pour déchiffrer une DITK dans le but de l'introduire dans le DecoderKeyRegister. De même, une DCTK ne doit pas être utilisée pour chiffrer une DITK dans le but de la transférer vers le compteur à paiement sous la forme d'un jeton.

Une DCTK peut également être utilisée pour chiffrer ou déchiffrer d'autres fonctions de gestion spécifiques à un compteur. Elle peut être utilisée pour chiffrer ou déchiffrer une fonction de transfert de crédit STS; autrement dit, un jeton de TransferCredit valide peut être chiffré ou déchiffré et appliqué par un compteur à paiement qui contient une DCTK dans son DKR et qui utilise un support de jeton de carte magnétique (valeur de TCT = 01).

6.5.2.4 Diagramme d'états des changements de DecoderKey



Légende

Anglais	Français
$DITK_0$	Clé de transfert d'initialisation de décodeur $DITK_0$
$DDTK_1$	Clé de transfert d'initialisation de décodeur $DITK_1$
$DUTK_2$	Clé de transfert d'initialisation de décodeur $DITK_2$
$DCTK_3$	Clé de transfert d'initialisation de décodeur $DITK_3$
Credit transfer functions	Fonctions de transfert de crédit
Management functions	Fonctions de gestion

Figure 10 – Changements de DecoderKey – diagramme d'états

La Figure 10 illustre les états de KT dans lesquels une DecoderKey peut être de temps en temps.

Lorsqu'une certaine clé est utilisée pour chiffrer une autre clé (comme dans la paire de jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey), la première est appelée "clé parente" et la seconde "clé enfant".

Les flèches en trait continu indiquent le sens dans lequel une clé peut passer d'un type à un autre type. Le type de départ est la clé parente et le type d'arrivée est la clé enfant. Pour effectuer un changement de la DecoderKey, la nouvelle clé (ou clé enfant) est chiffrée avec la clé parente et ensuite chargée dans le compteur à paiement au moyen de la paire de jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey. Le compteur à paiement remplace ensuite la clé parente par la clé enfant, qui devient maintenant la nouvelle clé parente.

Les flèches en pointillés indiquent la fonction, pour laquelle un KT peut être utilisé, c'est-à-dire les valeurs qu'il peut chiffrer ou déchiffrer. Par exemple, seule une DITK, une DUTK ou une DCTK peut être utilisée pour chiffrer ou déchiffrer une fonction de transfert de crédit, mais les quatre types peuvent être utilisés pour chiffrer ou déchiffrer des fonctions de gestion spécifiques à un compteur.

Le Tableau 26 donne dans le détail les relations autorisées des états de changement de clé et les fonctions associées.

Les lignes "clé enfant" renvoient à l'usage autorisé des types de clés de décodeur pour le chiffrement des DecoderKey dans les fonctions de gestion des jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey. De même, les lignes "gestion" et "crédit" donnent le détail de l'usage autorisé des types de clés de décodeur pour le chiffrement respectif des fonctions restantes de gestion et de transfert de crédit spécifiques à un compteur.

Tableau 26 – Relations autorisées entre les types de clés de décodeur

Clé enfant	Usage autorisé			
	Clé parente			
	$DITK_0$	$DDTK_1$	$DUTK_2$	$DCTK_3$
$DITK_0$	Oui	Non	Non	Non
$DDTK_1$	Oui	Oui	Oui	Oui[a]
$DUTK_2$	Oui	Oui	Oui	Oui[a]
$DCTK_3$	Oui[a]	Oui[a]	Non	Oui[a]
Fonction de gestion	Oui	Oui	Oui	Oui[a]
Fonction de crédit	Oui	Non	Oui	Oui[a]

[a] Pour les compteurs à paiement avec TCT = 01 seulement.

6.5.2.5 KeyRevisionNumber (KRN)

Un KRN est associé à chaque VendingKey et à un SGC correspondant par le KMS, et définit la révision ou la séquence de la VendingKey dans le SupplyGroup auquel il correspond. Il s'agit d'un chiffre décimal unique avec une plage de 1, 2..9. Le KRN assigné à la première VendingKey pour un SupplyGroup est 1. Les VendingKey qui suivent reçoivent des numéros de révision successifs qui leur sont alloués jusqu'au numéro de révision 9, auquel stade la séquence recommence à 1; autrement dit, à un moment donné quelconque, il ne peut pas y avoir plus de 9 révisions successives de la VendingKey présentes pour un SupplyGroup donné. Un KRN est également associé à chaque DecoderKey et correspond à celui de la VendingKey à partir de laquelle il est généré.

Le KRN est associé à chaque SupplyGroup par le KMS et définit la révision de VendingKey courante et également la révision de DecoderKey courante, auxquelles il convient d'établir tous les compteurs à paiement au sein du SupplyGroup. Pour tout compteur à paiement donné, le SGC et le KRN identifient de façon unique la révision de la DecoderKey qu'il contient. Cette information est gérée par le système de gestion et si, pour une raison quelconque, le KRN dans le compteur à paiement n'est pas le même que le KRN de vente pour le même SGC que celui enregistré dans le système de gestion, cette condition doit être corrigée au moyen d'un changement approprié de la DecoderKey.

Un compteur à paiement est tenu de stocker le KRN qui correspond à sa DecoderKey courante, telle que passée dans la paire de jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey (voir également 7.3.2).

Le concept de la révision de clé s'applique seulement aux types de clé de vente et aux types de clé de décodeur. Une DITK ne doit pas être associée à un KRN.

Pour un SupplyGroup donné, il doit y avoir un maximum de deux VendingKey actives dans le POS, à savoir la CurrentKey et l'OldKey. La clé OldKey est seulement utilisée pour chiffrer les

jetons de changement de clé en clé CurrentKey. La clé CurrentKey est utilisée pour chiffrer tous les jetons, à part les jetons de changement de clé, en clé OldKey

6.5.2.6 KeyExpiryNumber (KEN)

Un KEN est associé avec chaque VendingKey par le KMS et définit ce qui suit:

- la durée, après laquelle la VendingKey expire et ne peut plus être utilisée par un POS pour générer des DecoderKey dans le but de chiffrer les jetons de TransferCredit, ou les jetons de gestion spécifiques à un compteur qui incorporent le champ TID;
- la durée, après laquelle toute DecoderKey générée à partir de la VendingKey expire et ne peut plus être utilisée par un compteur à paiement pour accepter les jetons de TransferCredit, ou les jetons de gestion spécifiques à un compteur qui incorporent le champ TID. La mise en œuvre par le compteur à paiement est facultative.

La valeur exigée du KEN doit être respectivement transférée au compteur à paiement dans les champs KENHO et KENLO des jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey (voir 6.2.7 et 6.2.8).

Le KEN est un numéro de 8 bits (plage 0 – 255) qui exprime cette période comme un déplacement relatif au marqueur temporel de l'identificateur de jeton de date de référence STS (voir 6.3.5.1). Chaque unité dans le KEN correspond à une durée de 2^{16}-1 (65535) min et il y a 2^8 (256) de ces périodes numérotées 0,1..255 avant que le marqueur temporel courant de date de référence STS soit remplacé par le prochain marqueur temporel de référence STS. Le KEN correspond ainsi aux 8 bits de poids fort du TID de 24 bits. Aucun identificateur de jeton dont les 8 bits de poids fort sont supérieurs au KEN d'une clé donnée ne doit être chiffré ou déchiffré avec la clé en question.

Un POS ne peut pas émettre un jeton de TransferCredit chiffré sous une DecoderKey dont la VendingKey correspondante a expiré. Il est simple de le vérifier en comparant les 8 bits de poids fort du TID au KEN correspondant à la VendingKey; s'il est plus grand, la VendingKey a expiré et ne peut plus être utilisée pour générer une DecoderKey pour chiffrer le jeton de TransferCredit. Elle ne peut pas être également utilisée pour générer une DecoderKey pour chiffrer des jetons de gestion spécifiques à un compteur qui utilisent le champ TID. Cela ne s'applique pas à la paire de jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey qui n'utilisent pas le champ TID. Par conséquent, une DecoderKey expirée peut encore être utilisée pour chiffrer sa DecoderKey de remplacement dans le but d'un changement de DecoderKey.

Un compteur à paiement peut facultativement mettre en œuvre l'expiration de clé et stocker le KEN qui correspond à sa DecoderKey courante, tel que passé dans la paire de jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey. Tous les jetons qui sont introduits dans le compteur à paiement, et qui incorporent un champ identificateur de jeton, sont validés par rapport à ce KEN. Si les 8 bits de poids fort du TID sont supérieurs à ce KEN, le jeton doit être rejeté.

Lorsqu'il est mis en œuvre, le concept d'expiration de clé s'applique seulement aux valeurs de VendingKey du type VDDK, VUDK et VCDK, et aux valeurs de DecoderKey du type DDTK, DUTK et DCTK qui peuvent être générées à partir des types correspondants de clés de vente. Une DITK ne doit pas être associée à un KEN.

La gestion du KEN par le KMS doit être conforme au Code correspondant de bonnes pratiques.

Voir aussi C.2.4 pour le Code de bonnes pratiques de gestion de cet élément de données.

6.5.3 Génération de DecoderKey

6.5.3.1 Construction de PANBlock

Le PANBlock de 64 bits est construit à partir d'éléments de données extraits du MeterPAN dans l'APDU tels que définis dans le Tableau 27 et le Tableau 28.

Le chiffre de poids fort est à la position 15 et le chiffre de poids faible à la position 0.

Tableau 27 – Définition du PANBlock

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

Tableau 28 – Éléments de données dans le PANBlock

Chiffre	Nom	Format	Référence
I	IIN	Plage 0 à 9 hex par chiffre	6.1.2.2
D	DRN	Plage 0 à 9 hex par chiffre	6.1.2.3

Lorsque l'IIN a une longueur de 6 chiffres, le PANBlock est composé des 5 chiffres de poids faible de l'IIN et des 11 chiffres du DRN. Les 11 chiffres du DRN prennent les positions 10 à 0 dans le PANBlock et les 5 chiffres de poids faible de l'IIN prennent les positions 15 à 11 dans le PANBlock.

Lorsque l'IIN a une longueur de 4 chiffres, le PANBlock est composé des 3 chiffres de poids faible de l'IIN et des 13 chiffres du DRN. Les 13 chiffres du DRN prennent les positions 12 à 0 dans le PANBlock et les 3 chiffres de poids faible de l'IIN prennent les positions 15 à 13 dans le PANBlock.

Si l'IIN a une longueur insuffisante pour constituer les 16 chiffres, les chiffres extraits sont justifiés à droite au sein du bloc et complétés à gauche avec des zéros (par exemple: pour un IIN de 600727 et un DRN de 12345678903, le PANBlock est 0072712345678903).

Pour une DDTK ou DUTK, le DRN désigné réel est utilisé. Mais pour une DCTK, les chiffres de DRN sont mis à zéro dans le PANBlock (par exemple: pour un IIN de 600727, le PANBlock est 0072700000000000).

6.5.3.2 Construction de CONTROLBlock

Le CONTROLBlock de 64 bits est construit à partir des éléments de données dans l'APDU tels que définis dans le Tableau 29 et le Tableau 30.

Le chiffre de poids fort est à la position 15 et le chiffre de poids faible à la position 0.

Tableau 29 – Définition du CONTROLBlock

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

Tableau 30 – Éléments de données dans le CONTROLBlock

Chiffre	Nom	Format	Référence
C	Chiffre de KT	Plage 0 à 3 hex par chiffre, 4 à F hex = réservés	6.1.9
S	Chiffre de SGC	Plage 0 à 9 hex par chiffre	6.1.6
T	Chiffre de TariffIndex	Plage 0 à 9 hex par chiffre	6.1.7
R	Chiffre de KRN	Plage 1 à 9 hex par chiffre	6.1.8
F	Chiffre de valeur de Pad («bourrage»)	Toujours F hex par chiffre	x

6.5.3.3 DKGA01: DecoderKeyGenerationAlgorithm01

Ce DecoderKeyGenerationAlgorithm01 est à utiliser sur un petit ensemble limité de valeurs définies de DRN seulement. Il est inclus dans la présente Norme pour conserver la compatibilité aval avec un nombre limité de compteurs à paiement hérités conformes à la STS d'une génération antérieure utilisant également le STA (code EA 07). Le POSApplicationProcess donne la directive appropriée au moyen du code de DKGA dans l'APDU.

La DecoderKey est diversifiée à partir d'une valeur unique de VendingKey DES 64 bits.

Ce DecoderKeyGenerationAlgorithm01 est applicable à tous les compteurs à paiement qui satisfont à l'ensemble des critères suivants:

- utilisant IIN = 600727;
- et le KRN = 1;
- et le KT = 1 ou 2 (valeur par défaut ou unique);
- et le code EA 07 (STA);
- et le DRN s'inscrit dans les plages énumérées dans le Tableau 31.

Tableau 31 – Plage des valeurs applicables pour les numéros de référence de décodeur

Numéros de référence de décodeur		
0109000000X	à	0109000499X
0100000000X	à	0100499999X
0300000000X	à	0311400000X
0400000000X	à	0405999999X
0601000000X	à	0603999999X
0640000000X	à	0641999999X
0666000000X	à	0669999999X
0699000001X	à	0699000999X
0700000000X	à	0702099999X
NOTE X est un chiffre de contrôle, dont la valeur varie en fonction de la valeur des 10 chiffres précédents (voir 6.1.2.3)		

Ce DecoderKeyGenerationAlgorithm01 est également applicable à tous les compteurs à paiement qui satisfont à l'ensemble des critères suivants:

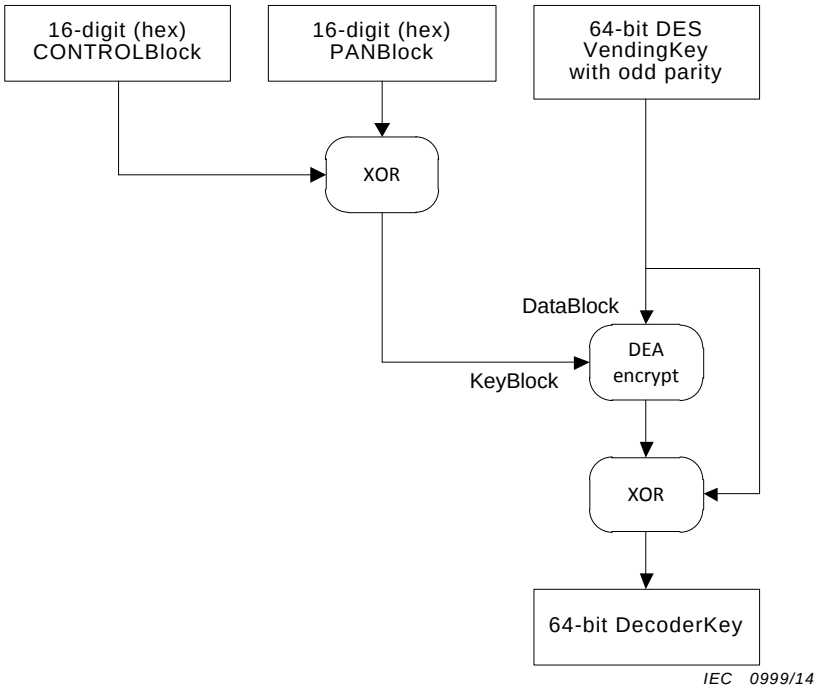
- utilisant IIN = 600727;
- et le KRN = 1;

- et le KT = 3 (commun);
- et le code EA 07 (STA);
- et codés avec l'une des valeurs de SGC énumérées dans le Tableau 32.

Tableau 32 – Liste des valeurs applicables pour les codes de groupe d'alimentation

Code de groupe d'alimentation
100702
990400
990401
990402
990403
990404
990405

Le flux de processus pour le DKGA01 est montré à la Figure 11.



Légende

Anglais	Français
16-digit (hex) CONTROLBlock	Bloc de contrôle de 16 chiffres (hex)
16-digit (hex) PANBlock	PANBlock de 16 chiffres (hex)
64-bit DES VendingKey with odd parity	Clé de vente DES 64 bits avec parité impaire
XOR	XOR («OU exclusif»)
DataBlock	Bloc de données
DEA encrypt	Chiffrer par le DEA
KeyBlock	Bloc de clés
64-bit DecoderKey	Clé de décodeur de 64 bits

Figure 11 – DecoderKeyGenerationAlgorithm01

Construire le PANBlock de 64 bits et le CONTROLBlock de 64 bits tels que définis en 6.5.3.1 et 6.5.3.2.

L'algorithme de chiffrement est DEA conformément au DES simple de la FIPS PUB 46-3 en mode ECB, utilisant une valeur unique de VendingKey DES de 64 bits avec parité impaire.

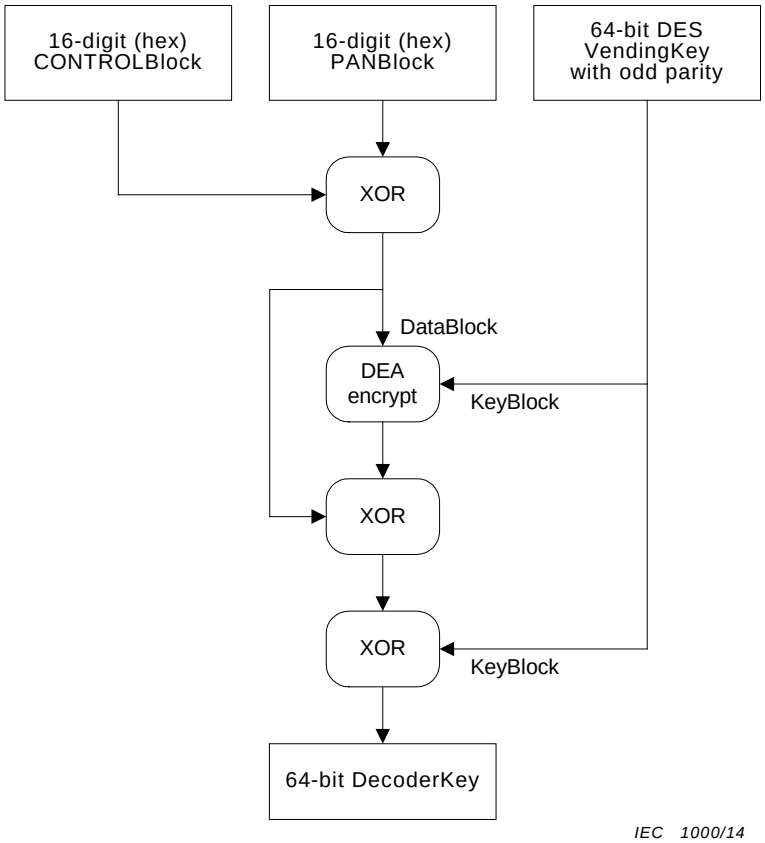
Dans ce cas, la clé VendingKey DES de 64 bits est utilisée comme l'entrée DataBlock conventionnelle au DEA, alors que le résultat de l'opération XOR du CONTROLBlock avec le PANBlock est utilisé comme l'entrée KeyBlock conventionnelle au DEA. Autrement dit, les blocs d'entrée de données et de clés sont permutés par rapport à la configuration conventionnelle.

6.5.3.4 DKGA02: DecoderKeyGenerationAlgorithm02

Le DecoderKeyGenerationAlgorithm02 peut être utilisé pour tous les compteurs à paiement qui ne satisfont pas aux critères de sélection du DecoderKeyGenerationAlgorithm01. Le POSApplicationProcess donne la directive appropriée au moyen du code de DKGA dans l'APDU.

La DecoderKey est diversifiée à partir d'une valeur unique de VendingKey DES 64 bits.

Le flux de processus pour le DKGA02 est montré à la Figure 12.



Légende

Anglais	Français
16-digit (hex) CONTROLBlock	Bloc de contrôle de 16 chiffres (hex)
16-digit (hex) PANBlock	PANBlock de 16 chiffres (hex)

Anglais	Français
64-bit DES VendingKey with odd parity	Clé de vente DES 64 bits avec parité impaire
XOR	XOR («OU exclusif»)
DataBlock	Bloc de données
DEA encrypt	Chiffrer par le DEA
KeyBlock	Bloc de clés
64-bit DecoderKey	Clé de décodeur de 64 bits

Figure 12 – DecoderKeyGenerationAlgorithm02

Construire le PANBlock de 64 bits et le CONTROLBlock de 64 bits tels que définis en 6.5.3.1 et 6.5.3.2.

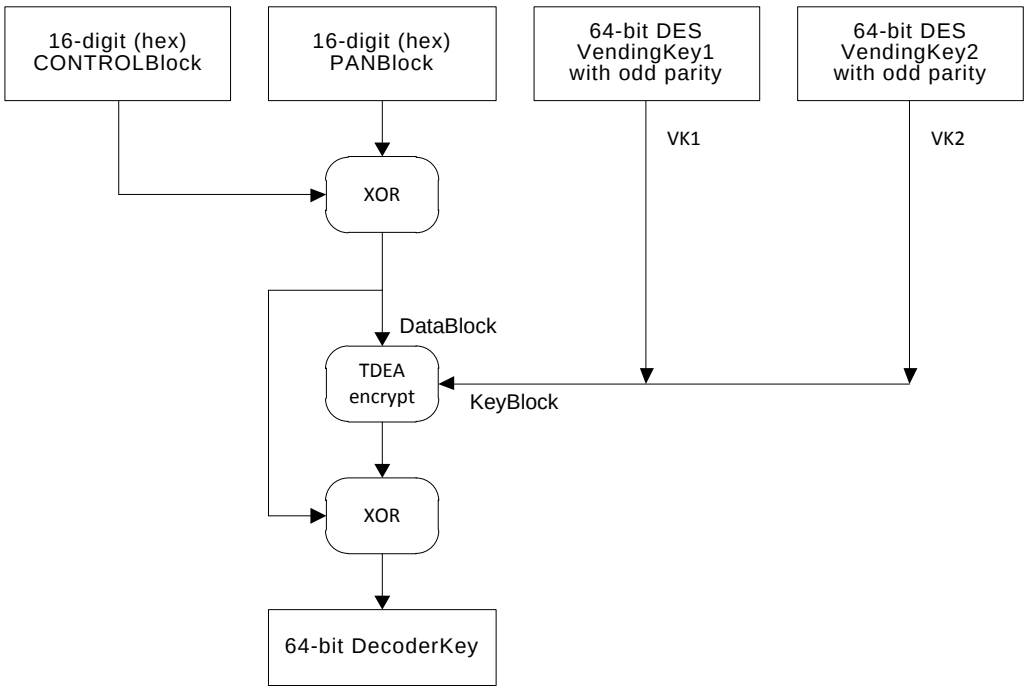
Le chiffrement est DEA conformément au DES simple de la FIPS PUB 46-3 en mode ECB, utilisant une valeur unique de VendingKey DES 64 bits avec parité impaire.

6.5.3.5 DKGA03: DecoderKeyGenerationAlgorithm03

Le DecoderKeyGenerationAlgorithm03 peut être utilisé pour tous les compteurs à paiement qui ne satisfont pas aux critères de sélection du DecoderKeyGenerationAlgorithm01. Le POSApplicationProcess donne la directive appropriée au moyen du code de DKGA dans l'APDU.

La DecoderKey est diversifiée à partir de deux valeurs de VendingKey DES 64 bits.

Le flux de processus pour le DKGA03 est montré à la Figure 13.



Légende

Anglais	Français
16-digit (hex) CONTROLBlock	Bloc de contrôle de 16 chiffres (hex)
16-digit (hex) PANBlock	PANBlock de 16 chiffres (hex)

Anglais	Français
64-bit DES VendingKey1 with odd parity	Clé de vente DES 64 bits n° 1 avec parité impaire
64-bit DES VendingKey2 with odd parity	Clé de vente DES 64 bits n° 2 avec parité impaire
VK1	VK1
VK2	VK2
XOR	XOR («OU exclusif»)
DataBlock	Bloc de données
TDEA encrypt	Chiffrer par le TDEA
KeyBlock	Bloc de clés
64-bit DecoderKey	Clé de décodeur de 64 bits

Figure 13 – DecoderKeyGenerationAlgorithm03

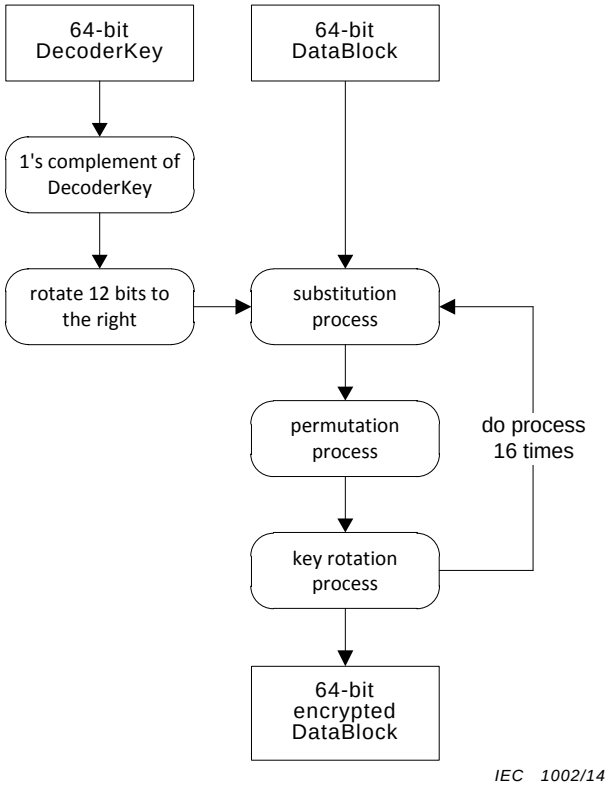
Construire le PANBlock de 64 bits et le CONTROLBlock de 64 bits tels que définis en 6.5.3.1 et 6.5.3.2.

Le chiffrement est TDEA conformément au DES triple de la FIPS PUB 46-3 en mode ECB, utilisant deux valeurs de VendingKey DES de 64 bits, à savoir VK1 et VK2, avec parité impaire.

L'opération est: chiffrer avec VK1, déchiffrer avec VK2, chiffrer avec VK1.

6.5.4 STA: EncryptionAlgorithm07

6.5.4.1 Processus de chiffrement



Légende

Anglais	Français
64-bit DecoderKey	Clé de décodeur de 64 bits

Anglais	Français
64-bit DataBlock	Bloc de données de 64 bits
1's complement of DecoderKey	complément à 1 de la clé de décodeur
rotate 12 bits to the right	tourner de 12 bits vers la droite
substitution process	processus de substitution
permutation process	processus de permutation
do process 16 times	exécuter le processus 16 fois
key rotation process	processus de rotation de clé
64-bit encrypted DataBlock	Bloc de données chiffré de 64 bits

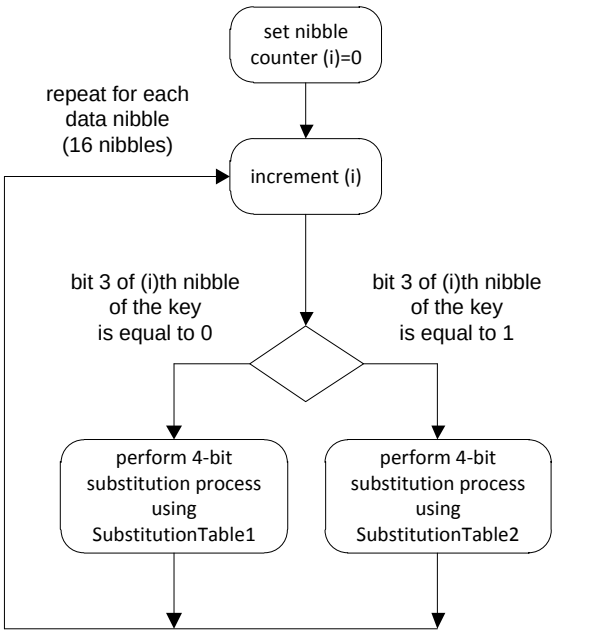
Figure 14 – STA: EncryptionAlgorithm07

Le processus de chiffrement par l'algorithme de transfert normalisé (Standard Transfer Algorithm) est montré à la Figure 14 et comprend un processus d'alignement de clé et 16 itérations d'un processus de substitution, de permutation et de rotation de clé.

Le POSApplicationProcess donne la directive appropriée au moyen du code EA dans l'APDU.

6.5.4.2 Processus de substitution

Le processus de substitution de chiffrement est illustré à la Figure 15.



Légende

Anglais	Français
set nibble counter (i)=0	mettre le compteur de quartets (i)=0
repeat for each data nibble(16 nibbles)	répéter pour chaque quartet de données (16 quartets)
increment (i)	incrémenter (i)
bit 3 of (i)th nibble of the key is equal to 0	le bit 3 du (i)^{ème} quartet de la clé est égal à 0
bit 3 of (i)th nibble of the key is equal to 1	le bit 3 du (i)^{ème} quartet de la clé est égal à 1
perform 4-bit substitution process using SubstitutionTable1	exécuter le processus de substitution de 4 bits en utilisant la SubstitutionTable1 (table de substitution n° 1)
perform 4-bit substitution process using	exécuter le processus de substitution de 4 bits en utilisant

Anglais	Français
SubstitutionTable2	la SubstitutionTable2 (table de substitution n° 2)

Figure 15 – Processus de substitution de chiffrement STA

Il y a un processus de substitution de 4 bits pour chacun des 16 quartets dans le train de données. La table de substitution utilisée est l'une de deux tables de substitution de 16 valeurs et dépend du positionnement du bit de poids fort du quartet correspondant dans la clé. Une table de substitution d'échantillons est donnée dans le Tableau 33.

Tableau 33 – Tables de substitution d'échantillons

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 Ce tableau ne contient que des valeurs d'échantillons (voir l'Article C.5 pour l'accès à une table avec des valeurs réelles).	

La première entrée de la table de substitution correspond à la position d'entrée 0 et la dernière à la position d'entrée 15.

Utiliser la valeur du quartet de données comme indice à une position d'entrée dans la table de substitution; remplacer ensuite la valeur de quartet par la valeur de la table de substitution qui se trouve à cette position d'entrée. Par exemple: si la valeur du quartet de données est 8 et la table utilisée est la SubstitutionTable1, alors l'entrée à la position 8 est la valeur 14. Remplacer alors la valeur de quartet de données par la valeur 14.

6.5.4.3 Processus de permutation

Le processus de permutation de chiffrement est illustré à la Figure 16.

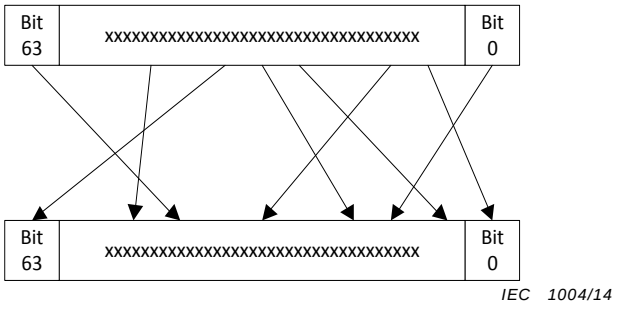


Figure 16 – Processus de permutation de chiffrement STA

Une table de permutation d'échantillons est donnée dans le Tableau 34.

Tableau 34 – Tableau de permutation d'échantillons

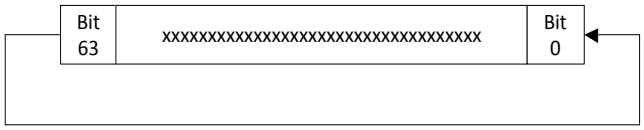
PermutationTable3	29, 27, 34, 9, 16, 62, 55, 2, 40, 49, 38, 25, 33, 61, 30, 23, 1, 41, 21, 57, 42, 15, 5, 58, 19, 53, 22, 17, 48, 28, 24, 39, 3, 60, 36, 14, 11, 52, 54, 12, 31, 51, 10, 26, 0, 45, 37, 43, 44, 6, 59, 4, 7, 35, 56, 50, 13, 18, 32, 47, 46, 63, 20, 8
NOTE Ce tableau ne contient que des valeurs d'échantillons (voir C.5 pour l'accès à une table avec des valeurs réelles).	

La première entrée dans la table de permutation correspond à la position 0 de bit de poids faible dans le DataBlock et la dernière entrée à la position 63 de bit de poids fort dans le DataBlock.

Utiliser la position binaire du DataBlock source comme indice dans la table de permutation; utiliser ensuite la valeur trouvée dans la table de permutation à cette position d'entrée comme un pointeur sur la position binaire dans le DataBlock de destination. Par exemple: pour le DataBlock source, la position binaire 7 correspond à la valeur 2 dans la table de permutation. La valeur de bit 7 du DataBlock source est donc mise à la position binaire 2 dans le DataBlock de destination.

6.5.4.4 Processus de rotation de clé

La clé entière est tournée d'une position binaire vers la gauche, comme montré à la Figure 17.

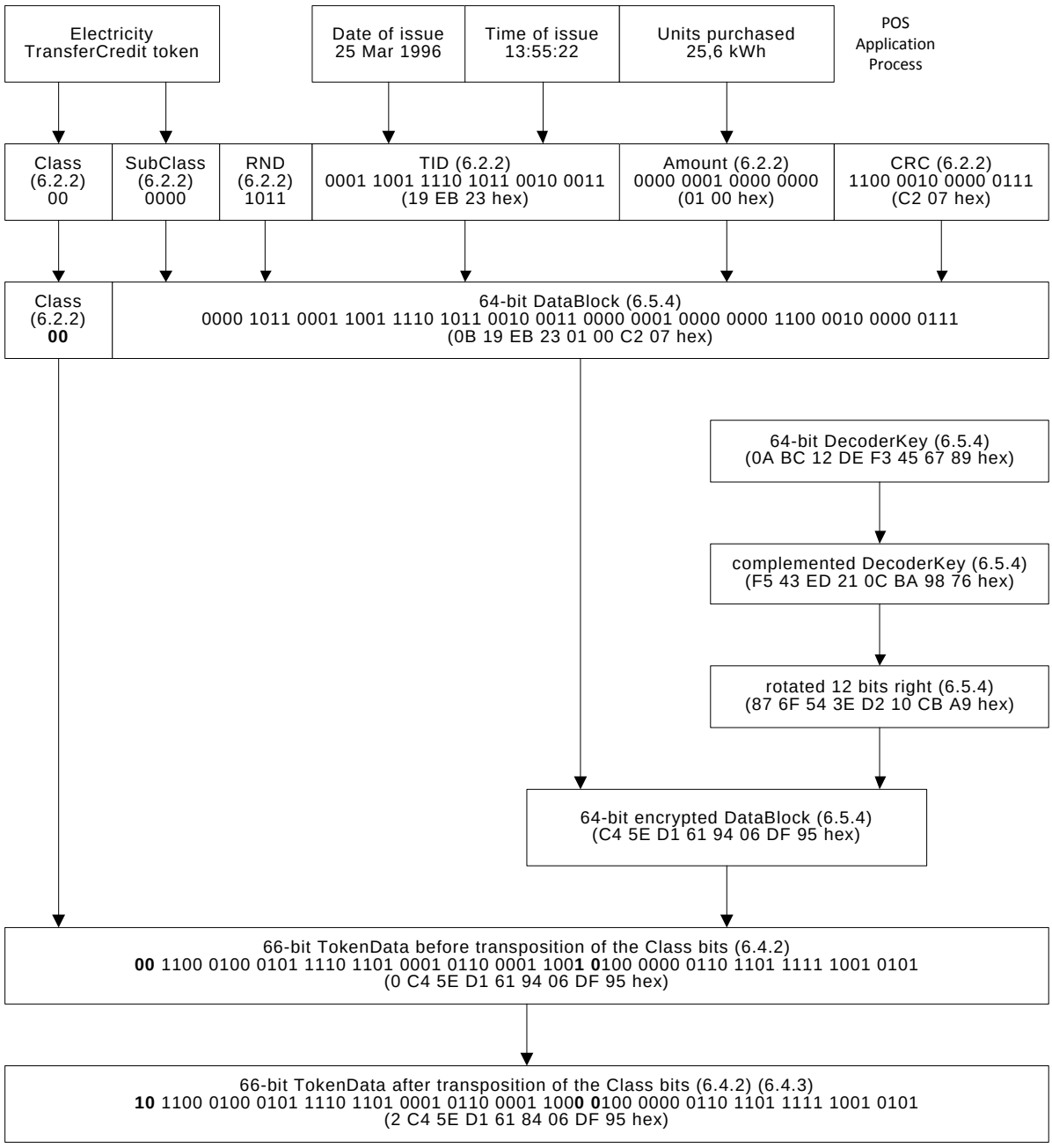


IEC 1005/14

Figure 17 – Processus de rotation de DecoderKey de chiffrement STA

6.5.4.5 Exemple pratique pour générer des TokenData pour un jeton de TransferCredit utilisant le STA

Un exemple pratique utilisant les tables de substitution et de permutation d'échantillons est illustré à la Figure 18.



IEC 1006/14

Légende

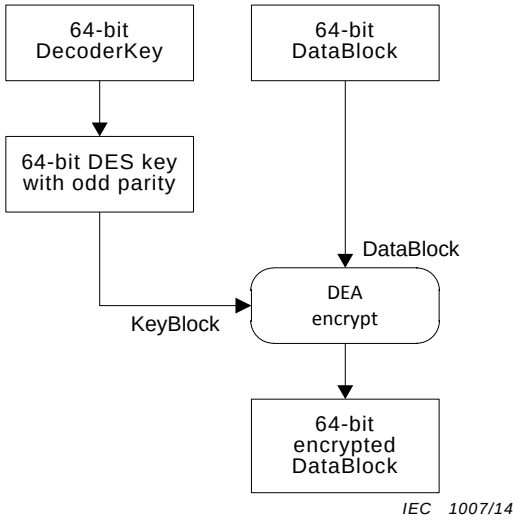
Anglais	Français
Electricity TransferCredit token	Jeton de crédit de transfert d'électricité
Date of issue 25 Mar 1996	Date d'émission 25 mars 1996
Time of issue 13:55:22	Heure d'émission 13:55:22
Units purchased 25,6 kWh	Unités achetées 25,6 kWh
RND (6.2.2) 1011	RND (6.2.2) 1011
CRC (6.2.2) 1100 0010 0000 0111 (C2 07 hex)	CRC (6.2.2) 1100 0010 0000 0111 (C2 07 hex)
Amount (6.2.2) 0000 0001 0000 0000 (01 00 hex)	Montant (6.2.2) 0000 0001 0000 0000 (01 00 hex)
TID (6.2.2) 0001 1001 1110 1011 0010 0011 (19 EB 23 hex)	TID (6.2.2) 0001 1001 1110 1011 0010 0011 (19 EB 23 hex)
Class (6.2.2) 00	Class («Classe») (6.2.2) 00

Anglais	Français
SubClass (6.2.2) 0000	SubClass («Sous-classe»)(6.2.2) 0000
64-bit DataBlock (6.5.4) 0000 1011 0001 1001 1110 1011 0010 0011 0000 0001 0000 0000 1100 0010 0000 0111 (0B 19 EB 23 01 00 C2 07 hex)	DataBlock («Bloc de données») de 64 bits (6.5.4) 0000 1011 0001 1001 1110 1011 0010 0011 0000 0001 0000 0000 1100 0010 0000 0111 (0B 19 EB 23 01 00 C2 07 hex)
64-bit DecoderKey (6.5.4) (0A BC 12 DE F3 45 67 89 hex)	DecoderKey («Clé de décodeur») de 64 bits (6.5.4) (0A BC 12 DE F3 45 67 89 hex)
complemented DecoderKey (6.5.4) (F5 43 ED 21 0C BA 98 76 hex)	DecoderKey («Clé de décodeur») complémentée (6.5.4) (F5 43 ED 21 0C BA 9876 hex)
rotated 12 bits right (6.5.4) (87 6F 54 3E D2 10 CB A9 hex)	tournée de 12 bits vers la droite (6.5.4) (87 6F 54 3E D2 10 CB A9 hex)
64-bit encrypted DataBlock (6.5.4) (C4 5E D1 61 94 06 DF 95 hex)	DataBlock («Bloc de données») chiffré de 64 bits (6.5.4) (C4 5E D1 61 94 06 DF 95 hex)
66-bit TokenData before transposition of the Class bits (6.4.2) **00**1100 0100 0101 1110 1101 0001 0110 0001 100**1 0**100 0000 0110 1101 1111 1001 0101 (0 C4 5E D1 61 94 06 DF 95 hex)	TokenData («Données du jeton») de 66 bits avant transposition des bits de Class (6.4.2) **00**1100 0100 0101 1110 1101 0001 0110 0001 100**1 0**100 0000 0110 1101 1111 1001 0101 (0 C4 5E D1 61 94 06 DF 95 hex)
POS Application Process	Processus application POS
66-bit TokenData after transposition of the Class bits (6.4.2) (6.4.3) **10**1100 0100 0101 1110 1101 0001 0110 0001 100**0 0**100 0000 0110 1101 1111 1001 0101 (2 C4 5E D1 61 84 06 DF 95 hex)	TokenData («Données du jeton») de 66 bits après transposition des bits de Class (6.4.2) (6.4.3) **10**1100 0100 0101 1110 1101 0001 0110 0001 100**0 0**100 0000 0110 1101 1111 1001 0101 (2 C4 5E D1 61 84 06 DF 95 hex)

Figure 18 – Exemple pratique de chiffrement STA pour un jeton de TransferCredit

6.5.5 DEA: EncryptionAlgorithm09

Le processus de chiffrement utilisant le DEA est montré à la Figure 19.



Légende

Anglais	Français
64-bit DecoderKey	Clé de décodeur de 64 bits
64-bit DataBlock	Bloc de données de 64 bits
64-bit DES key with odd parity	Clé DES 64 bits avec parité impaire
DataBlock	Bloc de données
KeyBlock	Bloc de clés
DEA encrypt	Chiffrer par le DEA
64-bit encrypted DataBlock	Bloc de données chiffré de 64 bits

Figure 19 – DEA: EncryptionAlgorithm09

Le DEA est un chiffrement bloc de 64 bits conforme à la FIPS PUB 46-3 opérant en mode ECB. Le POSApplicationProcess donne la directive appropriée au moyen du code EA dans l'APDU.

La DecoderKey de 64 bits est produite avec le DecoderKeyGenerationAlgorithm02 ou avec le DecoderKeyGenerationAlgorithm03 (voir 6.5.3.4 et 6.5.3.5).

La DecoderKey est convertie en clé DES 64 bits avec parité impaire conformément à la FIPS PUB 46-3 en changeant chaque huitième bit en bit de parité, en commençant par le bit de poids faible. Ainsi, le bit 0, le bit 8, le bit 16, le bit 24, le bit 32, le bit 40, le bit 48 et le bit 56 sont convertis en bits de parité, le bit 0 étant le bit de poids faible.

Le chiffrement est DEA conformément au DES simple de la FIPS PUB 46-3 en mode ECB, utilisant une valeur unique de clé DES 64 bits avec parité impaire.

7 Protocole de couche application de TokenCarriertoMeterInterface

7.1 APDU: ApplicationProtocolDataUnit

7.1.1 Éléments de données dans l'APDU

L'APDU, qui est l'interface de données entre le MeterApplicationProcess et le protocole de couche application, comprend les éléments de données indiqués dans le Tableau 35.

Tableau 35 – Éléments de données dans l'APDU

Élément	Contexte	Format	Référence
Token	Les TokenData issues de la TCDU après déchiffrement et traitement; maintenant présentées au MeterApplicationProcess dans l'APDU	66 bits	7.1.2
AuthenticationResult	Indicateur de statut au MeterApplicationProcess pour acheminer le résultat obtenu par des vérifications d'authentification initiales		7.1.3
ValidationResult	Indicateur de statut au MeterApplicationProcess pour acheminer le résultat obtenu par des vérifications de validation initiales		7.1.4
TokenResult	Indicateur de statut produit par le MeterApplicationProcess pour acheminer le résultat après traitement du jeton afin que le protocole de couche application puisse entreprendre l'action appropriée		7.1.5

7.1.2 Token

Les TokenData issues de la TCDU après déchiffrement et traitement; maintenant présentées au MeterApplicationProcess dans l'APDU.

Le jeton réel de 66 bits tel qu'introduit à l'origine dans l'APDU par le MeterApplicationProcess. Le MeterApplicationProcess est maintenant capable de lui donner un traitement supplémentaire. Voir 6.2.1 pour la définition détaillée de cet élément de données.

7.1.3 AuthenticationResult

Un indicateur de statut pour signaler au MeterApplicationProcess que les vérifications d'authentification initiales (voir 7.3.5) ont réussi ou échoué, afin que le MeterApplicationProcess puisse répondre de manière appropriée. Des valeurs possibles sont données dans le Tableau 36.

Tableau 36 – Valeurs possibles de l'AuthenticationResult

Valeur	Contexte	Format	Référence
Authentic	L'essai d'authentification a réussi ou échoué Faux si l'un quelconque des codes d'erreur ci-dessous est indiqué Vrai si aucun des codes d'erreur ci-dessous n'est indiqué	booléen	7.3.5
CRCError	La valeur de CRC dans le jeton est différente de la valeur de CRC calculée à partir des données contenues dans le jeton	booléen	7.3.5
MfrCodeError	La valeur de MfrCode dans le jeton de Class 1 ne concorde pas avec la valeur de MfrCode pour le Decoder	booléen	7.3.5

7.1.4 ValidationResult

Un indicateur de statut pour signaler au MeterApplicationProcess que les vérifications de validation initiales (voir 7.3.6) ont réussi ou échoué, afin que le MeterApplicationProcess

puisse répondre de manière appropriée. Des valeurs possibles sont données dans le Tableau 37.

Tableau 37 – Valeurs possibles du ValidationResult

Valeur	Contexte	Format	Référence
Valid	L'essai de Validation a réussi ou échoué Faux si l'un quelconque des codes d'erreur ci-dessous est indiqué Vrai si aucun des codes d'erreur ci-dessous n'est indiqué	booléen	7.3.6
OldError	La valeur de TID enregistrée dans le jeton est plus ancienne que la plus ancienne des valeurs enregistrées dans le stockage en mémoire du compteur à paiement	booléen	7.3.6
UsedError	La valeur de TID enregistrée dans le jeton est déjà enregistrée dans le stockage en mémoire du compteur à paiement	booléen	7.3.6
KeyExpiredError	La valeur de TID enregistrée dans le jeton est plus grande que le KEN enregistré dans la mémoire du compteur à paiement	booléen	7.3.6
DDTKError	Le Decoder a une valeur de DDTK dans le DKR; un jeton de TransferCredit ne peut pas être traité par le MeterApplicationProcess selon les règles données en 6.5.2.3.3.	booléen	7.3.6

7.1.5 TokenResult

Après que le MeterApplicationProcess a exécuté l'instruction contenue dans le jeton, la valeur de TokenResult reflète le résultat. Le protocole de couche application peut alors entreprendre l'action appropriée pour parachever le processus de lecture de jeton, qui peut inclure l'acceptation du jeton (et le stockage du TID), le rejet du jeton, l'effacement des données du jeton dans le TokenCarrier, etc. Des valeurs possibles sont données dans le Tableau 38.

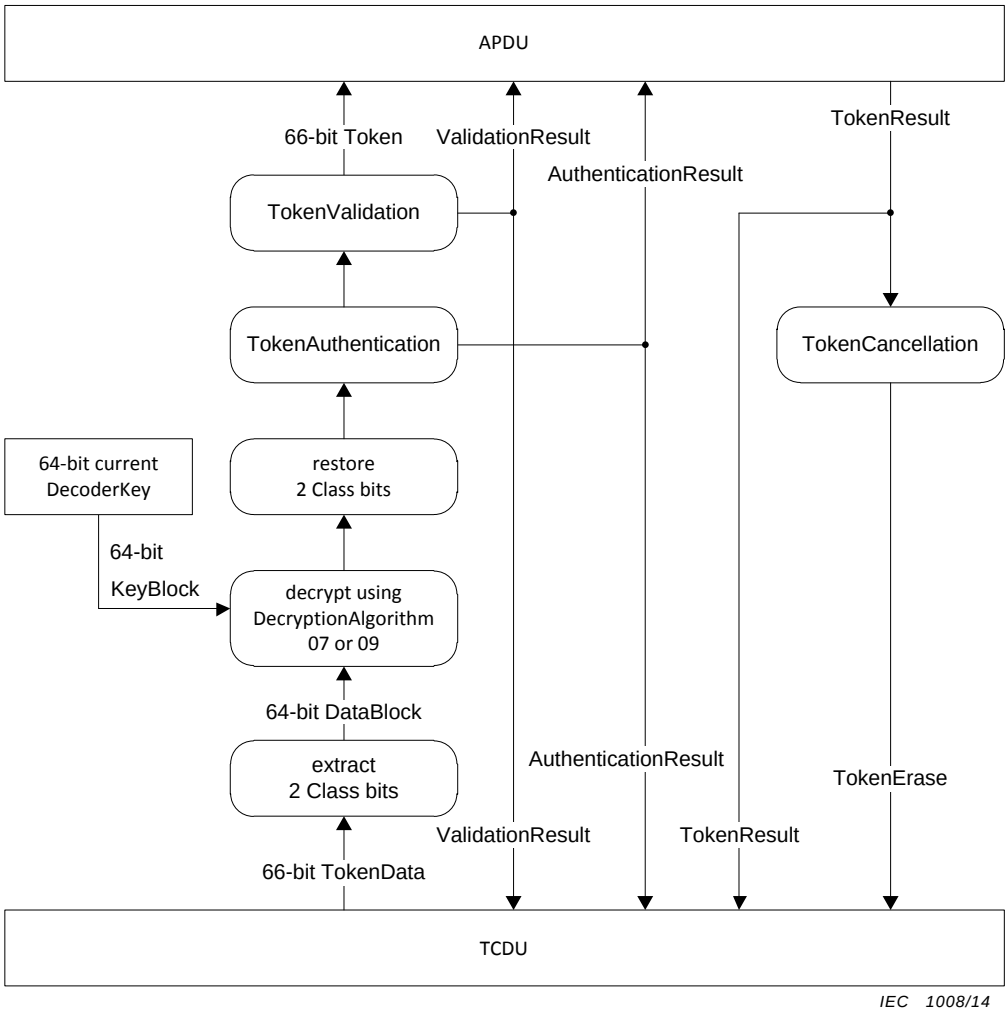
Tableau 38 – Valeurs possibles du TokenResult

Valeur	Contexte	Format	Référence
Accept	Le jeton a été traité avec succès Faux si l'un quelconque des codes d'erreur ci-dessous est indiqué Vrai si aucun des codes d'erreur ci-dessous n'est indiqué	booléen	8.2
1stKCT	Le MeterApplicationProcess indique qu'il s'agit du jeton Set1stSectionDecoderKey de la paire de jetons de changement de clé qui est lu; le jeton est accepté de façon provisoire	booléen	8.2
2ndKCT	Le MeterApplicationProcess indique qu'il s'agit du jeton Set2ndSectionDecoderKey de la paire de jetons de changement de clé qui est lu; le jeton est accepté de façon provisoire	booléen	8.2
OverflowError	Le registre de crédit dans le compteur à paiement déborde si le jeton est accepté; le jeton n'est pas accepté	booléen	8.2
KeyTypeError	La clé ne peut pas être changée en ce type selon les règles de changement de clé données en 6.5.2.4.	booléen	8.2
FormatError	Un ou plusieurs éléments de données dans le jeton ne sont pas conformes au format requis pour l'élément en question	booléen	8.2
RangeError	Un ou plusieurs éléments de données dans le jeton ont une valeur qui se situe à l'extérieur de la plage définie des valeurs définies dans l'application pour l'élément en question	booléen	6.3
FunctionError	La fonction particulière pour exécuter le jeton n'est pas mise en œuvre	booléen	8.2

7.2 Fonctions d'APDUExtraction

7.2.1 Processus d'extraction

Le processus d'extraction de l'APDU à partir de la TCDU est montré à la Figure 20.



Légende

Anglais	Français
APDU	APDU
66-bit Token	Jeton de 66 bits
ValidationResult	Résultat de validation
TokenResult	Résultat de jeton
AuthenticationResult	Résultat d'authentification
TokenValidation	Validation de jeton
TokenAuthentication	Authentification de jeton
TokenCancellation	Annulation de jeton
64-bit current DecoderKey	Clé de décodeur courante de 64 bits
restore 2 Class bits	restaurer les 2 bits de Class
64-bit KeyBlock	Bloc de clés de 64 bits
decrypt using DecryptionAlgorithm 07 or 09	déchiffrer en utilisant l'algorithme de déchiffrement DecryptionAlgorithm 07 ou 09
64-bit DataBlock	Bloc de données de 64 bits

Anglais	Français
extract 2 Class bits	extraire les 2 bits de Class
TokenErase	Effacer le jeton
66-bit TokenData	Données du jeton de 66 bits
TCDU	TCDU

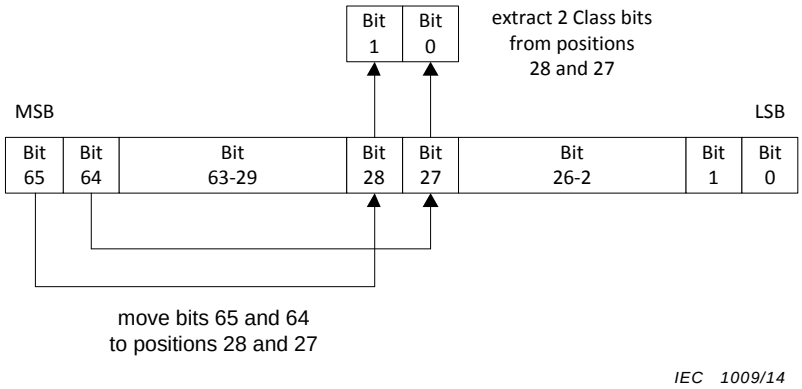
Figure 20 – Fonction d'APDUExtraction

La fonction APDUExtraction extrait les TokenData de 66 bits de la TCDU, les déchiffre et les traite avant de présenter le résultat dans l'APDU au MeterApplicationProcess. Elle annule finalement et fait facultativement effacer les données du jeton du TokenCarrier en réponse au résultat issu du MeterApplicationProcess.

7.2.2 Extraction des 2 bits de Class

Cette fonction est utilisée par d'autres fonctions d'APDUExtraction (voir 7.2.3 à 7.2.5). Elle retire les 2 bits de Class du train de données de 66 bits pour produire un nombre de 64 bits selon la méthode présentée dans les grandes lignes à la Figure 21 et elle est l'inverse de 6.4.2.

Le nombre de 66 bits a son bit de poids faible placé à la position binaire 0 et son bit de poids fort placé à la position binaire 65. La valeur de la Class de jetons de 2 bits est extraite des positions binaires 28 et 27. Les valeurs des positions binaires 65 et 64 sont déplacées aux positions binaires 28 et 27. Le bit de poids fort de la Class de jetons provient de la position binaire 28 d'origine.



Légende

Anglais	Français
extract 2 Class bits from positions 28 and 27	extraire des positions 28 et 27 les 2 bits de Class
MSB	Bit de poids fort
LSB	Bit de poids faible
move bits 65 and 64 to positions 28 and 27	déplacer vers les positions 28 et 27 les bits 65 et 64

Figure 21 – Extraction des 2 bits de Class

Exemple: Extraction de la Class de jetons = 01 (binaire)

Extraire les 2 bits de Class des positions binaires 28 et 27 (en gras):

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

Déplacer les bits 65 et 64 bits vers les positions binaires 28 et 27 (en gras):

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

Le nombre binaire résultant de 64 bits groupé en quartets (Les bits 27 et 28 sont signalés en gras):

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

7.2.3 Fonction APDUExtraction pour les jetons de Class 0 et Class 2

Il s'agit de la fonction de transfert de la TCDU vers l'APDU et elle s'applique à tous les jetons de Class 0 et Class 2, excepté les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey (voir 7.2.5).

NOTE 1 Les éléments de données dans l'APDU sont définis en 7.1.1.

NOTE 2 Les éléments de données dans la TCDU sont définis dans chaque partie de la norme de protocole de couche physique de la série IEC 62055-5x applicable au TCT spécifique d'intérêt.

La fonction de transfert pour les jetons de Class 0 et Class 2 est présentée dans les grandes lignes comme suit:

- les 2 bits de Class sont extraits des TokenData de 66 bits en utilisant la méthode en 7.2.2 pour donner un résultat de 64 bits, qui est alors présenté à l'algorithme de déchiffrement comme entrée de DataBlock. Noter qu'il est de la responsabilité du POS d'enregistrer quel algorithme particulier de déchiffrement est utilisé dans chaque compteur à paiement particulier (voir 6.1.5 EA). L'algorithme de déchiffrement et l'algorithme de chiffrement sont complémentaires et partagent donc le même code EA;
- l'entrée KeyBlock pour l'algorithme de déchiffrement contient la valeur courante de la DecoderKey, qui est obtenue à partir du DecoderKeyRegister dans la mémoire sécurisée du compteur à paiement;
- après déchiffrement, les 2 bits de Class sont de nouveau réinsérés dans le nombre de 64 bits pour produire un nombre de 66 bits. Le bit de poids fort des 2 bits de Class entre dans la position binaire 65 et le bit de poids faible de Class entre dans la position binaire 64;
- le jeton de 66 bits est authentifié selon 7.3.5 et le résultat est indiqué dans le champ AuthenticationResult de l'APDU;
- le jeton de 66 bits est validé selon 7.3.6 et le résultat est indiqué dans le champ ValidationResult de l'APDU et le jeton de 66 bits est mis dans le champ Token de l'APDU;
- le MeterApplicationProcess traite le Token de l'APDU et indique le résultat dans le champ TokenResult de l'APDU (voir également 8.2). Il est de la responsabilité du MeterApplicationProcess de traiter des messages et des indicateurs d'affichage (voir également 8.3) à l'attention de l'utilisateur et pas celle du protocole de couche application;
- si le TokenResult indique Accept (voir 7.1.5 et 8.2), le Token est annulé selon 7.3.7 et l'instruction est donnée dans le champ TokenErase de la TCDU d'effacer les données du TokenCarrier.

NOTE 3 Il est de la responsabilité du protocole de couche physique de décider si, oui ou non, l'instruction d'effacement est applicable conformément à sa mise en œuvre spécifique et au TCT (voir l'Article 6 de l'IEC 62055-51:2007, par exemple).

7.2.4 Fonction APDUExtraction pour les jetons de Class 1

La fonction APDUExtraction pour les jetons de Class 1 est identique à celle des jetons de Class 0 et de Class 2, excepté que l'étape de déchiffrement n'est pas exécutée.

7.2.5 Fonction APDUExtraction pour les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey

Il s'agit de la fonction de transfert de la TCDU vers l'APDU et elle est applicable aux jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey.

NOTE 1 Les éléments de données dans l'APDU sont définis en 7.1.1.

NOTE 2 Les éléments de données dans la TCDU sont définis dans chaque partie de la norme de protocole de couche physique de la série IEC 62055-5x applicable au TCT spécifique d'intérêt.

La fonction de transfert pour les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey est présentée dans les grandes lignes comme suit:

- les 2 bits de Class sont extraits des TokenData de 66 bits en utilisant la méthode en 7.2.2 pour donner un résultat de 64 bits, qui est alors présenté à l'algorithme de déchiffrement comme entrée de DataBlock. Noter qu'il est de la responsabilité du POS d'enregistrer quel algorithme particulier de déchiffrement est utilisé dans chaque compteur à paiement particulier (voir 6.1.5 EA). L'algorithme de déchiffrement et l'algorithme de chiffrement sont complémentaires et partagent donc le même code EA;
- l'entrée KeyBlock pour l'algorithme de déchiffrement contient la valeur courante de la DecoderKey, qui est obtenue à partir du DecoderKeyRegister dans la mémoire sécurisée du compteur à paiement;
- après déchiffrement, les 2 bits de Class sont de nouveau réinsérés dans le nombre de 64 bits pour produire un nombre de 66 bits. Le bit de poids fort des 2 bits de Class entre dans la position binaire 65 et le bit de poids faible de Class entre dans la position binaire 64;
- le jeton de 66 bits est authentifié selon 7.3.5 et le résultat est indiqué dans le champ AuthenticationResult de l'APDU;
- le jeton de 66 bits n'est pas validé dans le protocole de couche application, il l'est seulement dans le MeterApplicationProcess. Le jeton de 66 bits est mis dans le champ Token de l'APDU;
- le MeterApplicationProcess traite le Token de l'APDU et indique le résultat dans le champ TokenResult de l'APDU (voir également 8.2). Il est de la responsabilité du MeterApplicationProcess de traiter des messages et des indicateurs d'affichage (voir également 8.3) à l'attention de l'utilisateur et pas celle du protocole de couche application;
- si le TokenResult indique 1stKCT ou 2ndKCT (voir 7.1.5 et 8.2), l'instruction d'effacer les données du TokenCarrier n'est pas donnée dans le champ TokenErase de la TCDU;
- si le TokenResult indique Accept (voir 7.1.5 et 8.2), l'instruction d'effacer les données du TokenCarrier est donnée dans le champ TokenErase de la TCDU.

Les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey peuvent être introduits dans un ordre quelconque (voir 8.9), mais seul le dernier doit être effacé.

NOTE 3 Il est de la responsabilité du protocole de couche physique de décider si, oui ou non, l'instruction d'effacement est applicable conformément à sa mise en œuvre spécifique et au TCT (voir l'Article 6 de l'IEC 62055-51:2007, par exemple).

7.3 Fonctions de sécurité

7.3.1 Attributs de clé et changements de clé

7.3.1.1 Exigences relatives au changement de clé

Le compteur à paiement doit se conformer aux exigences pertinentes de 6.5.2, 7.3.1.2 et 7.3.1.3.

7.3.1.2 Traitement de changement de clé sans expiration de clé

Ce qui suit définit le traitement de changement de clé requis si l'expiration de clé n'est pas mise en œuvre dans le compteur à paiement:

– comparer la valeur de KT sur le jeton avec la valeur de KT dans le compteur à paiement:
  - si les valeurs de KT sont égales, changer le contenu du DecoderKeyRegister, le KRN du décodeur et le TI du compteur à paiement dans les nouvelles valeurs correspondantes sur le jeton;

- si les valeurs de KT ne sont pas égales, valider les règles de KT (voir 6.5.2.4):
  a) si le changement de clé est autorisé, changer le contenu du DecoderKeyRegister, le KRN du décodeur, le KT du décodeur et le TI du compteur à paiement dans les nouvelles valeurs correspondantes sur le jeton;
  b) si le changement de clé n'est pas autorisé, rejeter l'opération de changement de clé.

7.3.1.3 Traitement de changement de clé avec expiration de clé

Ce qui suit définit le traitement de changement de clé requis si l'expiration de clé est mise en œuvre dans le compteur à paiement:

– comparer la valeur de KT de jeton avec la valeur KT de décodeur:
  - si les valeurs de KT sont égales, changer le contenu du DecoderKeyRegister, le KEN du décodeur, le KRN du décodeur et le TI du compteur à paiement dans les valeurs correspondantes de jeton;
  - si les valeurs de KT ne sont pas égales, valider les règles de KT (voir 6.5.2.4):
    a) si le changement de clé est autorisé, changer le contenu du DecoderKeyRegister, le KEN de décodeur, le KRN de décodeur, le KT de décodeur et le TI du compteur à paiement dans les valeurs correspondantes de jeton;
    b) si le changement de clé n'est pas autorisé, rejeter l'opération de changement de clé.

7.3.2 DKR: DecoderKeyRegister

Le compteur à paiement doit stocker les valeurs données dans le Tableau 39 dans une mémoire sécurisée non volatile.

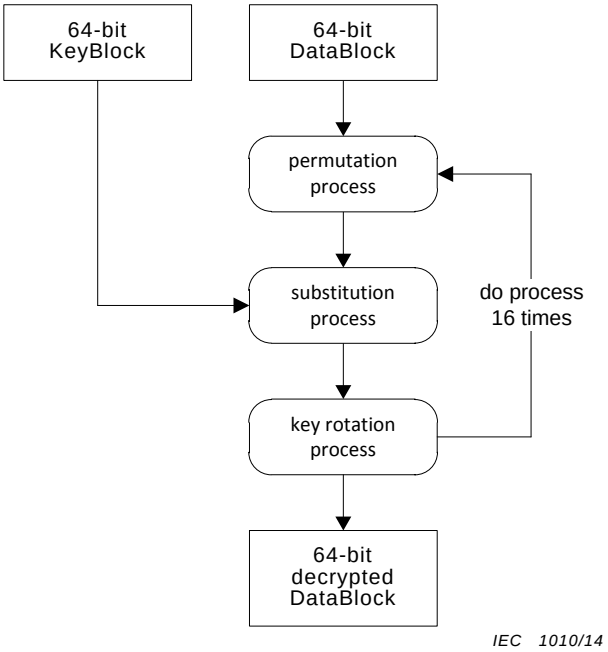
Tableau 39 – Valeurs stockées dans le DKR

Valeur	Référence
DecoderKey (Clé de décodeur)	6.5.2.3.3, 6.5.3
TI	6.1.7
KRN	6.1.8
KT	6.1.9
KEN (facultatif)	6.1.10
Le TI peut être associé à une table Tariff qui est gérée à l'extérieur du domaine du compteur à paiement. Cela implique que si une entreprise de distribution se sert de l'association, le compteur à paiement exige un changement de clé chaque fois que le consommateur est associé à une structure tarifaire différente.	

Dans tous les cas où le compteur à paiement fournit des informations de configuration, le KT doit être considéré comme étant une partie des informations de KeyRevisionNumber. Le compteur à paiement doit donc toujours fournir les informations relatives au KT avec, ou autrement directement après, les informations relatives au KRN.

7.3.3 STA: DecryptionAlgorithm07

7.3.3.1 Processus de déchiffrement



Légende

Anglais	Français
64-bit KeyBlock	Bloc de clés de 64 bits
64-bit DataBlock	Bloc de données de 64 bits
permutation process	processus de permutation
substitution process	processus de substitution
do process 16 times	exécuter le processus 16 fois
key rotation process	processus de rotation de clé
64-bit decrypted DataBlock	Bloc de données déchiffré de 64 bits

Figure 22 – DecryptionAlgorithm07 STA

Le processus de déchiffrement par l'algorithme de transfert normalisé (Standard Transfer Algorithm) est montré à la Figure 22 et comprend un processus d'alignement de clé et 16 itérations d'un processus de permutation, de substitution et de rotation de clé.

L'algorithme de déchiffrement et l'algorithme de chiffrement sont complémentaires et partagent donc le même code EA.

7.3.3.2 Processus de permutation

Le processus de permutation de déchiffrement est illustré à la Figure 23.

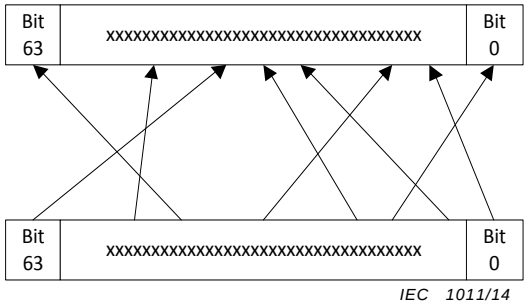


Figure 23 – Processus de permutation de déchiffrement STA

Une table de permutation d'échantillons est donnée dans le Tableau 40.

Tableau 40 – Tableau de permutation d'échantillons

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 Ce tableau ne contient que des valeurs d'échantillons (voirl'Article C.5 pour l'accès à une table avec des valeurs réelles).	

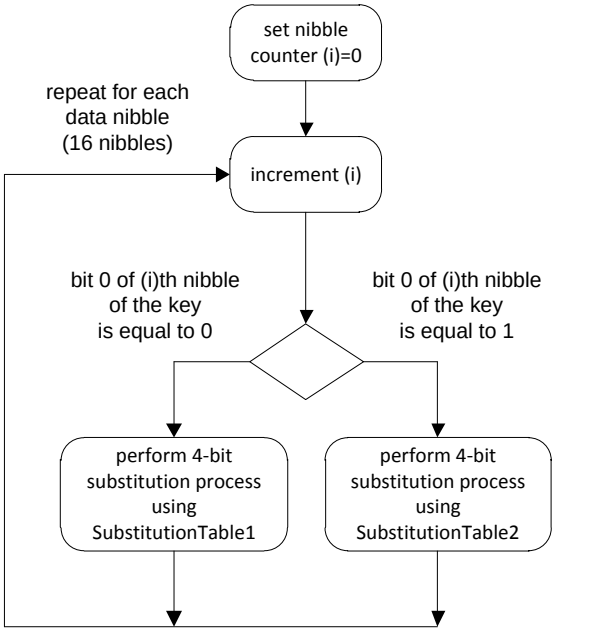
La première entrée dans la table de permutation correspond à la position 0 de bit de poids faible dans le DataBlock et la dernière entrée à la position 63 de bit de poids fort dans le DataBlock.

Utiliser la position binaire du DataBlock source comme indice dans la table de permutation; utiliser ensuite la valeur trouvée dans la table de permutation à cette position d'entrée comme un pointeur sur la position binaire dans le DataBlock de destination. Par exemple: pour le DataBlock source, la position binaire 7 correspond à la valeur 52 dans la table de permutation. La valeur de bit 7 du DataBlock source est donc mise à la position binaire 52 dans le DataBlock de destination.

Cela donne visiblement le résultat inverse de celui du processus en 6.5.4.3.

7.3.3.3 Processus de substitution

Le processus de substitution de déchiffrement est illustré à la Figure 24.



Légende

Anglais	Français
set nibble counter (i)=0	mettre le compteur de quartets (i)=0
repeat for each data nibble(16 nibbles)	répéter pour chaque quartet de données (16 quartets)
increment (i)	incrémenter (i)
bit 0 of (i)th nibbleof the key is equal to 0	le bit 0 du (i)^{ème} quartet de la clé est égal à 0
bit 0 of (i)th nibbleof the key is equal to 1	le bit 0 du (i)^{ème} quartet de la clé est égal à 1
perform 4-bit substitution process using SubstitutionTable1	exécuter le processus de substitution de 4 bits en utilisant la SubstitutionTable1 (table de substitution n° 1)
perform 4-bit substitution process using SubstitutionTable2	exécuter le processus de substitution de 4 bits en utilisant la SubstitutionTable1 (table de substitution n° 2)

Figure 24 – Processus de substitution de déchiffrement STA

Il y a un processus de substitution de 4 bits pour chacun des 16 quartets dans le train de données. La table de substitution utilisée est l'une de deux tables de substitution de 16 valeurs et dépend du positionnement du bit de poids faible du quartet correspondant dans la clé. Une table de substitution d'échantillons est donnée dans le Tableau 41.

Tableau 41 – Tables de substitution d'échantillons

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 Ce tableau ne contient que des valeurs d'échantillons (voir l'Article C.5 pour l'accès à une table avec des valeurs réelles).	

La première entrée de la table de substitution correspond à la position d'entrée 0 et la dernière à la position d'entrée 15.

Utiliser la valeur du quartet de données comme indice à une position d'entrée dans la table de substitution; remplacer ensuite la valeur de quartet par la valeur de la table de substitution qui se trouve à cette position d'entrée. Par exemple: si la valeur du quartet de données est 8 et la table utilisée est la SubstitutionTable1, alors l'entrée à la position 8 est la valeur 14. Remplacer alors la valeur de quartet de données par la valeur 14.

Cela donne visiblement le résultat inverse de celui du processus en 6.5.4.2.

7.3.3.4 Processus de rotation de clé

La clé entière est tournée d'une position binaire vers la droite, comme montré à la Figure 25.

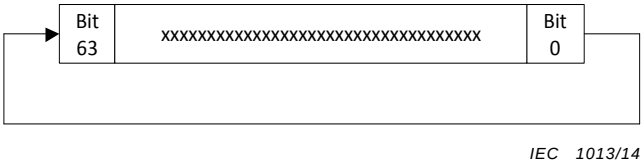
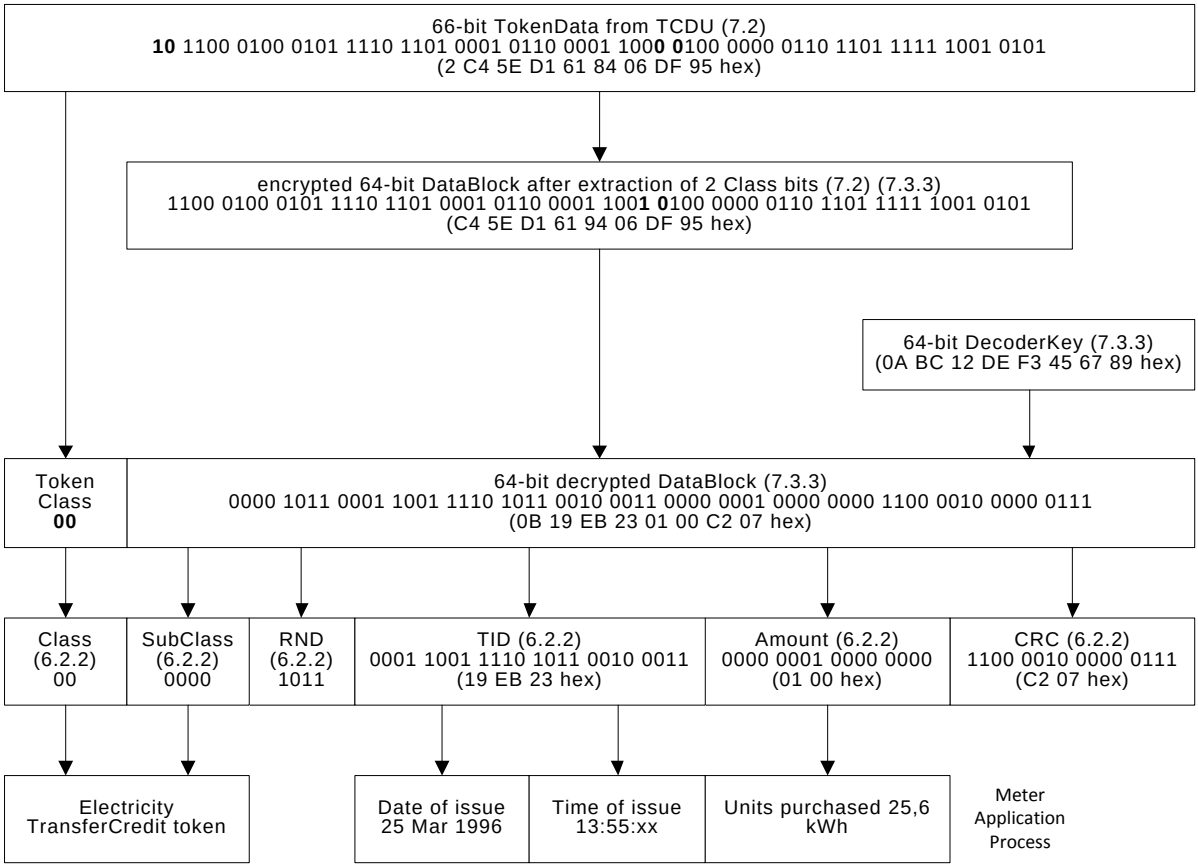


Figure 25 – Processus de rotation de DecoderKey de déchiffrement STA

7.3.3.5 Exemple pratique pour déchiffrer un jeton TransferCredit en utilisant le STA

Un exemple pratique utilisant les tables de substitution et de permutation d'échantillons est illustré à la Figure 26.



IEC 1014/14

Légende

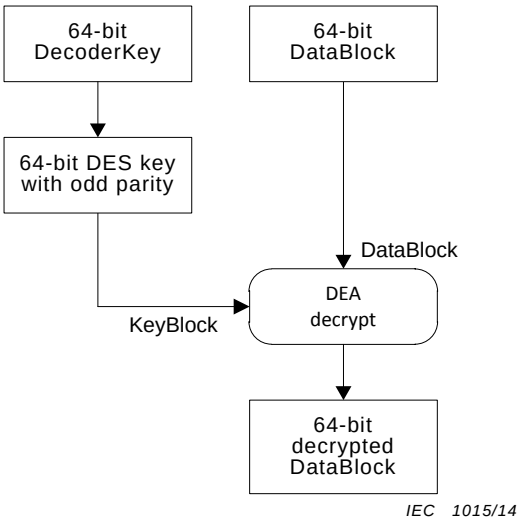
Anglais	Français
66-bit TokenData from TCDU (7.2) **10**1100 0100 0101 1110 1101 0001 0110 0001 100**0 0**100 0000 0110 1101 1111 1001 0101 (2 C4 5E D1 61 84 06 DF 95 hex)	TokenData («données du jeton») de 66 bits issues de la TCDU (7.2) **10**1100 0100 0101 1110 1101 0001 0110 0001 100**0 0**100 0000 0110 1101 1111 1001 0101 (2 C4 5E D1 61 84 06 DF 95 hex)
encrypted 64-bit DataBlock after extraction of 2 Class bits (7.2) (7.3.3) 1100 0100 0101 1110 1101 0001 0110 0001 100**1 0**100 0000 0110 1101 1111 1001 0101 (C4 5E D1 61 94 06 DF 95 hex)	DataBlock («Bloc de données») de 64 bits chiffré après extraction des 2 bits de classe (7.2) (7.3.3) 1100 0100 0101 1110 1101 0001 0110 0001 100**1 0**100 0000 0110 1101 1111 1001 0101 (C4 5E D1 61 94 06 DF 95 hex)
64-bit DecoderKey (7.3.3) (0A BC 12 DE F3 45 67 89 hex)	DecoderKey («Clé de décodeur») de 64 bits (7.3.3) (0A BC 12 DE F3 45 67 89 hex)
Token Class **00**	Token Class («Classe de jetons») **00**
64-bit decrypted DataBlock (7.3.3) 0000 1011 0001 1001 1110 1011 0010 0011 0000 0001 0000 0000 1100 0010 0000 0111 (0B 19 EB 23 01 00 C2 07 hex)	DataBlock («bloc de données») déchiffré de 64 bits (7.3.3) 0000 1011 0001 1001 1110 1011 0010 0011 0000 0001 0000 0000 1100 0010 0000 0111 (0B 19 EB 23 01 00 C2 07 hex)
Class (6.2.2) 00	Class («Classe») (6.2.2) 00
SubClass (6.2.2) 0000	SubClass («Sous-classe») (6.2.2) 0000
RND (6.2.2) 1011	RND (6.2.2) 1011
TID (6.2.2) 0001 1001 1110 1011 0010 0011 (19 EB 23 hex)	TID (6.2.2) 0001 1001 1110 1011 0010 0011 (19 EB 23 hex)
Amount (6.2.2) 0000 0001 0000 0000 (01 00 hex)	Montant (6.2.2) 0000 0001 0000 0000 (01 00 hex)

Anglais	Français
CRC (6.2.2) 1100 0010 0000 0111 (C2 07 hex)	CRC (6.2.2) 1100 0010 0000 0111 (C2 07 hex)
Electricity TransferCredit token	Jeton de crédit de transfert d'électricité
Date of issue 25 Mar 1996	Date d'émission 25 mars 1996
Time of issue 13:55:xx	Heure d'émission 13:55:xx
Units purchased 25,6 kWh	Unités achetées 25,6 kWh
Meter Application Process	Processus d'application compteur

Figure 26 – Exemple pratique de déchiffrement STA pour un jeton de TransferCredit

7.3.4 DEA: DecryptionAlgorithm09

Le processus de déchiffrement utilisant le DEA est montré à la Figure 27.



Légende

Anglais	Français
64-bit DecoderKey	Clé de décodeur de 64 bits
64-bit DataBlock	Bloc de données de 64 bits
64-bit DES key with odd parity	Clé DES 64 bits avec parité impaire
DataBlock	Bloc de données
DEA decrypt	déchiffrer par le DEA
KeyBlock	Bloc de clés
64-bit decrypted DataBlock	Bloc de données déchiffré de 64 bits

Figure 27 – DEA DecryptionAlgorithm09

Le DEA est un chiffrement bloc de 64 bits conforme à la FIPS PUB 46-3 opérant en mode ECB.

La DecoderKey est convertie en clé DES 64 bits avec parité impaire conformément à la FIPS PUB 46-3 en changeant chaque huitième bit en bit de parité, en commençant par le bit de poids faible. Ainsi, le bit 0, le bit 8, le bit 16, le bit 24, le bit 32, le bit 40, le bit 48 et le bit 56 sont convertis en bits de parité, le bit 0 étant le bit de poids faible.

L'algorithme de déchiffrement et l'algorithme de chiffrement sont complémentaires et partagent donc le même code EA.

Le déchiffrement est DEA conformément au DES simple de la FIPS PUB 46-3 en mode ECB, utilisant une valeur unique de clé DES 64 bits avec parité impaire.

7.3.5 TokenAuthentication

La validation de la somme de contrôle CRC après déchiffrement doit authentifier les jetons de Class 0 et de Class 2.

La validation du CRC et du MfrCode doit authentifier les jetons de Class 1.

Dans le cas d'un jeton de Class 0 ou de Class 2, le statut d'AuthenticationResult doit indiquer Authentic lorsque la condition suivante est remplie:

- la somme de contrôle de CRC dans le jeton a la même valeur que celle qui est calculée à partir des éléments de données dans le jeton.

Si la condition ci-dessus n'est pas remplie, le statut d'AuthenticationResult doit indiquer CRCError.

Dans le cas d'un jeton de Class 1, le statut d'AuthenticationResult doit indiquer Authentic lorsque toutes les deux conditions suivantes sont remplies:

- la somme de contrôle de CRC dans le jeton a la même valeur que celle qui est calculée à partir des éléments de données dans le jeton;
- la valeur du MfrCode dans le jeton est la même que celle du MfrCode défini en 6.2.3.

Si l'une quelconque des conditions ci-dessus n'est pas remplie, le statut d'AuthenticationResult doit indiquer CRCError et/ou MfrCodeError.

Si le jeton ne peut pas être authentifié, il doit être rejeté conformément à l'exigence donnée en 8.2 et 8.3.

7.3.6 TokenValidation

Les jetons de Class 0 et de Class 2 doivent principalement être validés par rapport au TID codé dans le jeton, excepté les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey.

Les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey sont validés par le MeterApplicationProcess une fois que le compteur à paiement a lu les deux jetons et les a combinés en la nouvelle DecoderKey. Voir 8.2 pour les exigences relatives à l'acceptation et au rejet des jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey.

Si l'expiration de clé est mise en œuvre dans le compteur à paiement, le KEN stocké dans le compteur à paiement doit être également utilisé pour valider les jetons de Class 0 et de Class 2 (voir 6.5.2.6), excepté les jetons de Set1stSectionDecoderKey et Set2ndSectionDecoderKey.

Un statut de "Valid" doit être indiqué si aucune des conditions suivantes n'est vraie:

- Si un TID est reçu et a une valeur plus petite que la plus petite valeur de TID stocké dans le stockage en mémoire (autrement dit, il a été émis par un POS à une date antérieure au TID le plus précoce stocké dans le stockage en mémoire), un tel jeton contenant ce TID doit être rejeté et indiquer une telle condition sous la forme d'un statut OldError (voir 7.1.4);
- Si un TID est reçu et a déjà été stocké dans le stockage en mémoire (voir 7.3.7), le jeton doit être rejeté et indiquer une telle condition sous la forme d'un statut UsedError (voir 7.1.4);

- Si l'expiration de clé est mise en œuvre dans le compteur à paiement et un TID est reçu en étant supérieur au KEN dans le Decoder, le jeton doit être rejeté et indiquer une telle condition sous la forme d'un statut KeyExpiredError (voir 7.1.4);
- Si un jeton de Class 0 est présenté au Decoder avec une valeur de DDTK dans le DKR, le jeton doit être rejeté (voir 6.5.2.3.3) et indiquer une telle condition sous la forme d'un statut DDTKError (voir 7.1.4).

Voir aussi 8.2 et 8.3 pour les exigences relatives à l'acceptation, au rejet et à l'indication du MeterApplicationProcess.

Un compteur à paiement chargé avec une valeur de DDTK doit accepter tous les "jetons de gestion non spécifiques à un compteur" (jetons de Class 1) pertinents ainsi que les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey chiffrés sous une DDTK.

7.3.7 TokenCancellation

L'annulation d'un jeton doit être effectuée au moyen du stockage du TID associé à ce jeton dans le stockage en mémoire sécurisée non volatile en plus de l'effacement de l'enregistrement des données du jeton dans les supports de jetons de cartes magnétiques (voir 6.1.3 et 6.2.5 de l'IEC 62055-51:2007).

Un TID basé sur le temps est utilisé pour identifier de façon unique chaque jeton de Class 0 et de Class 2 (excepté les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey). Le compteur à paiement doit stocker, dans un stockage en mémoire sécurisée non volatile, au moins les 50 dernières valeurs de TID reçues.

Si un jeton valide est reçu avec un TID qui a une valeur plus grande que la plus petite des valeurs de TID dans le stockage en mémoire et il n'y a pas d'espace disponible dans le stockage en mémoire pour stocker la valeur de TID reçue, le compteur à paiement doit accepter ce jeton, retirer la plus petite valeur de TID (autrement dit, le TID le plus ancien) du stockage en mémoire, et la remplacer par la nouvelle valeur de TID.

Si le compteur à paiement accepte une paire de jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey, le stockage en mémoire du TID doit rester inchangé, à moins que le champ RolloverKeyChange (voir 6.3.18) ne spécifie que le stockage en mémoire doit être vidé.

Le compteur à paiement ne doit pas accepter des jetons qui avaient été créés avant la date de fabrication ou de réparation du compteur à paiement.

NOTE Une méthode conseillée est que le constructeur remplisse le stockage en mémoire de TID avec des valeurs qui indiquent la date et l'heure de fabrication ou de réparation.

Le compteur à paiement doit lire et traiter un jeton (et également l'effacer si requis) sur une seule insertion du TokenCarrier sans autre action de l'utilisateur.

Tous les compteurs à paiement fonctionnant avec une DCTK (voir 6.5.2.3.1) doivent effacer les données du jeton (jeton de Class 0 et de Class 2) du TokenCarrier après le transfert réussi des données du jeton du TokenCarrier vers le compteur à paiement, excepté les données du jeton Set1stSectionDecoderKey et les données du jeton Set2ndSectionDecoderKey.

Les jetons suivants ne doivent pas être effacés:

- tout jeton portant un TID qui est jugé vieux par le compteur à paiement;
- les "jetons de gestion non spécifiques à un compteur" de Class 1;
- le jeton Set1stSectionDecoderKey ou un Set2ndSectionDecoderKey, celui de ces deux jetons qui est inséré le premier étant choisi.

Le jeton Set1stSectionDecoderKey ou un jeton Set2ndSectionDecoderKey, celui de ces deux jetons qui est inséré le dernier étant choisi, doit être effacé à la suite d'un achèvement réussi de l'opération de changement de clé.

8 Exigences du MeterApplicationProcess

8.1 Exigences générales

En plus des exigences données à l'Article 8, le MeterApplicationProcess doit exécuter les jetons conformes aux définitions données à l'Article 6 et à l'Article 7 et doit en plus être soumis aux exigences données dans l'IEC 62055-31, et ce, à tout moment, notamment l'action du commutateur de charge en réponse au réapprovisionnement à distance du crédit et la fermeture du commutateur de charge à partir d'un emplacement distant.

8.2 Acceptation / rejet de jeton

Un compteur à paiement conforme à la STS doit être capable de lire, interpréter et exécuter avec succès toutes les catégories de jetons.

Le compteur à paiement doit encore accepter des jetons lorsqu'il est dans l'état de limitation de puissance ou dans l'état falsifié.

Les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey sont validés par le MeterApplicationProcess une fois que le compteur à paiement a lu les deux jetons et les a combinés en la nouvelle DecoderKey.

Un jeton doit être accepté lorsque toutes les conditions suivantes sont vraies:

- AuthenticationResult indique une valeur de statut "Authentic" dans l'APDU (voir 7.1.3);
- ValidationResult indique une valeur de statut "Valid" dans l'APDU (voir 7.1.4);
- le jeton peut être interprété correctement et l'instruction exécutée par le MeterApplicationProcess.

Si toutes les conditions ci-dessus sont remplies, TokenResult (voir 7.1.5) doit indiquer "Accept" avec les exceptions suivantes:

- le traitement réussi du premier jeton introduit d'une paire de jetons de changement de clé ne doit pas indiquer "Accept", mais il doit indiquer "1stKCT" s'il s'agit d'un jeton Set1stSectionDecoderKey ou "2ndKCT" s'il s'agit d'un jeton Set2ndSectionDecoderKey; cela indique une acceptation provisoire jusqu'à ce que le second jeton de la paire de jetons de changement de clé soit également accepté;
- le traitement réussi du second jeton introduit d'une paire de jetons de changement de clé doit indiquer "Accept".

Le jeton doit être rejeté et le TokenResult ne doit pas indiquer "Accept" si l'une quelconque des conditions suivantes est vraie:

- AuthenticationResult n'indique pas une valeur de statut "Authentic" dans l'APDU (voir 7.1.3);
- AuthenticationResult indique une valeur de statut "CRCError" dans l'APDU (voir 7.1.3);
- AuthenticationResult indique une valeur de statut "MfrCodeError" dans l'APDU (voir 7.1.3);
- ValidationResult n'indique pas une valeur de statut "Valid" dans l'APDU (voir 7.1.4);
- ValidationResult indique une valeur de statut "OldError" dans l'APDU (voir 7.1.4);
- ValidationResult indique une valeur de statut "UsedError" dans l'APDU (voir 7.1.4);
- ValidationResult indique une valeur de statut "KeyExpiredError" dans l'APDU (voir 7.1.4);

- ValidationResult indique une valeur de statut "DDTKError" dans l'APDU (voir 7.1.4);
- Au cas où le parachèvement de l'exécution de transaction d'un jeton TransferCredit fait déborder le registre de crédit dans le compteur à paiement, le TokenResult doit indiquer OverflowError dans l'APDU (voir 7.1.5) au lieu de "Accept", le jeton doit être rejeté et ne doit pas être traité davantage;
- Au cas où l'exécution d'un jeton de changement de clé viole les règles de changement de clé données en 6.5.2.4, le TokenResult doit indiquer KeyTypeError dans l'APDU (voir 7.1.5) au lieu de "Accept", le jeton doit être rejeté et ne doit pas être traité davantage. Voir aussi 7.3.1 pour d'autres exigences relatives au traitement de changement de clé;
- Au cas où la structure du jeton ne se conforme pas aux définitions données en 6.2, 6.3 ou dans l'application pour le jeton en question, le TokenResult doit indiquer "FormatError" dans l'APDU (voir 7.1.5) au lieu de "Accept", le jeton doit être rejeté et ne doit pas être traité davantage;
- Au cas où un ou plusieurs éléments de données dans le jeton ont une valeur qui se situe à l'extérieur de la plage définie de valeurs définie en 6.2, 6.3 ou dans l'application pour l'élément en question, le TokenResult doit indiquer "RangeError" dans l'APDU (voir 7.1.5) au lieu de "Accept", le jeton doit être rejeté et ne doit pas être traité davantage;
- Au cas où la fonction particulière pour exécuter le jeton n'est pas mise en œuvre, le TokenResult doit indiquer "FunctionError" dans l'APDU (voir 7.1.5) au lieu de "Accept", le jeton doit être rejeté et ne doit pas être traité davantage.

8.3 Indicateurs d'affichage et marquages

Le compteur à paiement doit indiquer de manière unique les conditions suivantes:

- l'acceptation d'un jeton (voir 8.2);
- le rejet d'un jeton (voir 8.2);
- lorsqu'un jeton est vieux (voir 7.1.4);
- lorsqu'un jeton a déjà été utilisé, c'est-à-dire un jeton en doublon (voir 7.1.4);
- lorsque la DecoderKey a expiré (voir 7.1.4);
- lorsqu'un jeton de TransferCredit est présenté avec une DDTK dans le DKR (voir 7.1.4);
- lorsque le MeterApplicationProcess ne peut pas exécuter le jeton (voir 8.2);
- après l'achèvement réussi d'une opération de changement de clé (voir 8.2 et 8.9);
- si, oui ou non, l'acceptation du crédit sur un jeton fait déborder le registre de crédit (voir 8.2).

Le DRN et le code EA doivent être marqués sur la partie du compteur à paiement qui contient la partie "décodeur" (voir Article 3) et doivent être lisibles de l'extérieur du décodeur.

Au cas où la partie de décodeur est séparée de l'interface du TokenCarrier où l'utilisateur présente le TokenCarrier au compteur à paiement, il doit être possible pour l'utilisateur de déterminer le DRN et le code EA à partir de l'interface utilisateur, sur demande.

Les indicateurs relatifs au résultat d'entrée de jeton ne doivent être affichés que sur la même interface utilisateur où le jeton a été introduit. Dans le cas d'un support de jeton virtuel par exemple, il incombe au protocole de couche application et au protocole de couche physique approprié la tâche d'alimenter en retour les valeurs de ValidationResult, d'AuthenticationResult et de TokenResult.

8.4 Jetons de TransferCredit

Voir 6.2.2 pour plus de détails relatifs à la structure de ce jeton.

La valeur de crédit dans le champ Amount dans le jeton doit être ajoutée au crédit disponible dans la fonction Accounting selon la mise en œuvre spécifique de la fonction Accounting et le type de service indiqué par le champ Subclass dans le jeton.

8.5 Jetons InitiateMeterTest/Display

Voir 6.2.3 pour plus de détails relatifs à la structure de ce jeton.

Tous les compteurs à paiement doivent prendre en charge l'essai numéro 0; si l'un ou plusieurs des essais incorporés ne sont pas pris en charge, le compteur à paiement doit effectuer le sous-ensemble des essais qui sont pris en charge.

L'essai approprié doit être réalisé ou l'information appropriée doit être affichée selon le profil binaire dans le champ Control du jeton.

Lorsque plus d'une sortie est requise, par exemple pour l'essai numéro 0, les sorties doivent être initiées dans l'ordre dans lequel elles sont définies en 6.3.8. Un essai facultatif peut être omis s'il n'est pas mis en œuvre. Un seul et même essai (l'essai numéro 3, par exemple) peut fournir plus d'un champ d'informations.

Tous les essais facultatifs qui ne sont pas pris en charge par le compteur à paiement doivent conduire au rejet du jeton d'essai facultatif par le compteur à paiement.

Dans le cas où la valeur de SubClass se situe dans la plage 6 à 15, l'essai approprié ou la fonction d'affichage appropriée doit être exécuté(e) selon la spécification du constructeur, mais le compteur à paiement doit vérifier la valeur du champ MfrCode avant qu'un tel jeton ne soit accepté.

Dans le cas où un compteur à paiement a un crédit disponible de zéro, ce qui fait ouvrir le commutateur de charge, et le jeton InitiateMeterTest/Display peut faire fonctionner le commutateur de charge à l'état fermé pendant la durée de l'essai. Certaines entreprises de distribution peuvent ne pas vouloir que cet état soit autorisé, alors que d'autres entreprises de distribution peuvent le vouloir. L'action du compteur à paiement en réponse à ce jeton doit être telle que convenue entre l'entreprise de distribution et le fournisseur et ne doit pas constituer une partie normative de la présente Norme.

8.6 Jetons SetMaximumPowerLimit

Voir 6.2.4 pour plus de détails relatifs à la structure de ce jeton.

La présente valeur du registre de limite maximale de puissance doit être remplacée par la nouvelle limite.

L'action de cette fonction doit être convenue entre l'entreprise de distribution et le fournisseur de compteur à paiement.

NOTE 1 Dans un compteur à paiement à plusieurs phases, cette valeur est par phase.

NOTE 2 Cette fonction n'est pas destinée à être utilisée comme un mécanisme de protection contre les surintensités, qui exige l'adhésion à d'autres normes pertinentes.

8.7 Jetons ClearCredit

Voir 6.2.5 pour plus de détails relatifs à la structure de ce jeton.

Le crédit disponible dans la fonction Accounting doit être effacé à zéro selon la valeur indiquée dans le champ Register (Registre) du jeton.

8.8 Jetons SetTariffRate

Voir 6.2.6 pour plus de détails relatifs à la structure de ce jeton.

La présente valeur dans le Tariff Rate Register (registre des taux de tarif) doit être remplacée par le nouveau taux.

8.9 Jetons Set1stSectionDecoderKey

Voir 6.2.7 pour plus de détails relatifs à la structure de ce jeton.

La présente valeur de la DecoderKey doit être remplacée par la nouvelle DecoderKey. La DecoderKey inclut ses attributs associés comme KRN, KT, KEN et TI définis en 7.3.2.

Cette action est assujettie à la réception réussie des deux jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey.

Le compteur à paiement doit avoir une seule DecoderKey active à un stade quelconque de son fonctionnement. Les DecoderKey doubles ne doivent pas être utilisées.

Il doit être possible d'introduire les jetons Set1stSectionDecoderKey et Set2ndSection DecoderKey dans n'importe quel ordre pour réaliser un changement de clé réussi.

Il doit être possible d'introduire au moins deux autres jetons non valides de tout type et dans n'importe quel ordre, avec n'importe lequel des jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey, et accomplir tout de même un changement réussi de clé.

Il doit être possible d'introduire plus d'une fois le même jeton Set1stSectionDecoderKey ou Set2ndSectionDecoderKey, si la clé n'a pas déjà été changée, et accomplir tout de même un changement réussi de clé.

Une fonction de temporisation doit être utilisée pour annuler au bout d'une durée comprise entre 3 min et 10 min une procédure de changement de clé partiellement aboutie.

8.10 Jetons Set2ndSectionDecoderKey

Voir 6.2.8 pour plus de détails relatifs à la structure de ce jeton.

Les exigences relatives au traitement des jetons Set2ndSectionDecoderKey sont les mêmes qu'en 8.9 ci-dessus.

8.11 Jetons ClearTamperCondition

Voir 6.2.9 pour plus de détails relatifs à la structure de ce jeton.

Le statut Control et l'indicateur qui signale un état de fraude doivent être réinitialisés pour indiquer un état d'absence de fraude. Tout processus interne de commande de compteur à paiement découlant d'un tel état de fraude doit aussi être annulé.

8.12 Jetons SetMaximumPhasePowerUnbalanceLimit

Voir 6.2.10 pour plus de détails relatifs à la structure de ce jeton.

La présente valeur du registre de limite maximale de déséquilibre de puissance de phases doit être remplacée par la nouvelle limite.

L'action de cette fonction doit être convenue entre l'entreprise de distribution et le fournisseur de compteur à paiement.

NOTE Cette fonction ne s'applique qu'aux compteurs à paiement à plusieurs phases.

8.13 SetWaterMeterFactor

Voir 6.2.11 pour plus de détails relatifs à la structure de ce jeton.

L'action de ce jeton est réservée pour une définition future par la STS Association.

8.14 Classe 2: Jetons réservés pour l'usage selon la STS

Voir 6.2.12 pour plus de détails relatifs à la structure de ce jeton.

Le compteur à paiement doit rejeter ces types de jetons.

8.15 Classe 2: Jetons réservés pour un usage propriétaire

Voir 6.2.13 pour plus de détails relatifs à la structure de ce jeton.

Les actions accomplies dans le compteur à paiement doivent être conformes aux spécifications du constructeur.

NOTE La présente Norme n'assure pas la protection contre une collision entre usages constructeur de cet espace de jetons.

8.16 Classe 3: Jetons réservés pour l'usage selon la STS

Voir 6.2.14 pour plus de détails relatifs à la structure de ce jeton.

Le compteur à paiement doit rejeter ces types de jetons.

9 KMS: Exigences génériques relatives au KeyManagementSystem

Il est reconnu que les exigences relatives au KMS ne relèvent essentiellement pas du domaine d'application de la présente Norme. Le lecteur est donc renvoyé aux normes appropriées de l'industrie, dont certaines sont énumérées dans la Bibliographie.

La STS Association a établi des codes de bonnes pratiques bien éprouvés pour la gestion des clés cryptographiques au sein des systèmes conformes à la STS, en utilisant les normes de l'industrie en question. La recommandation est donc qu'il convient que les nouveaux systèmes mettant en œuvre la présente Norme suivent les codes de bonnes pratiques de la STS Association.

En vertu de son statut d'Autorité d'enregistrement auprès du CE 13 de l'IEC, la STS Association a entrepris de fournir de tels services de certification qui sont considérés comme étant nécessaires pour assurer que les systèmes de gestion de clé sont conformes aux parties pertinentes de la présente Norme (voir l'Article C.1). Pour des lignes directrices supplémentaires relatives au fonctionnement d'un KeyManagementSystem tel qu'envisagé dans la présente Norme, voir l'Annexe A.

10 Maintenance des entités STS et services connexes

10.1 Généralités

Voir aussi l'Article C.1 pour plus d'informations relatives aux services de maintenance et d'assistance.

L'activité maintenance sur certaines entités STS exige une révision/un amendement de la présente Norme. Lorsque tel est le cas, cela est indiqué de façon explicite comme tel.

L'Annexe B et l'Annexe C ne sont pas normatives et tout changement dans ces Articles en raison d'activités de maintenance n'exige pas une révision/un amendement de la présente Norme, mais peut exiger des amendements appropriés à d'autres spécifications ou Codes de bonnes pratiques (COP) pertinent(e)s.

Les entités et services STS qui exigent une maintenance sont donnés dans le Tableau 42.

Tableau 42 – Entités/services exigeant un service de maintenance

Entité/service	Origine de la définition	Organisme de maintenance responsable	Référence
Certification de produit	Article C.10	STSA/CA	10.2.1
DSN	6.1.2.3.3 C.3.4	Mfr	10.2.2
RO	6.3.18	entreprise de distribution	10.2.3
TI	6.1.7	entreprise de distribution	10.2.4
TID	6.3.5.1	entreprise de distribution	10.2.5
SpecialReservedTokenIdentifier	6.3.5.2 Article C.4	entreprise de distribution	10.2.6
MfrCode	6.1.2.3.2 C.3.3	STSA	10.2.7
Tables de substitution	6.5.4.2 7.3.3.3 Article C.5	STSA	10.2.8
Tables de permutation	6.5.4.3 7.3.3.2 Article C.5	STSA	10.2.9
SGC	6.1.6 C.2.2	STSA/KMC	10.2.10
VendingKey (Clé de vente)	6.5.2.2 Article 9 C.2.2	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 Tableau 30	STSA/KMC	10.2.13
KEN	6.1.10 6.5.2.6 C.2.4	STSA/KMC	10.2.14
KEK	Annexe B Tableau B.1	STSA/KMC	10.2.15
CC	Annexe B Tableau B.2	STSA/KMC	10.2.16
UC	Annexe B Tableau B.2	STSA/KMC	10.2.17
KMCID	Annexe B Tableau B.2	STSA/KMC	10.2.18
CMID	Annexe B Tableau B.2	Mfr/KMC	10.2.19
CMAC	Annexe B Tableau B.2	Mfr/KMC	10.2.20

Entité/service	Origine de la définition	Organisme de maintenance responsable	Référence
IIN	6.1.2.2 C.3.2	ISO/IEC	10.3.1
TCT	6.1.3 Tableau 5	STSA/IEC	10.3.2
DKGA	6.1.4 Tableau 6	STSA/IEC	10.3.3
EA	6.1.5 Tableau 7	STSA/IEC	10.3.4
TokenClass	6.3.2 Tableau 13 Tableau 14	STSA/IEC	10.3.5
TokenSubClass	6.3.3 Tableau 14	STSA/IEC	10.3.6
InitiateMeterTest/DisplayControlField	6.3.8 Tableau 22	STSA/IEC	10.3.7
RegisterToClear	6.3.13 Tableau 23	STSA/IEC	10.3.8
STS base date (c'est-à-dire: Date de référence STS)	6.3.5.1	STSA/IEC	10.3.9
Rate (c'est-à-dire: Taux)	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	Article C.8	STSA/(IEC)	10.3.13
Companion Specification (c'est-à-dire: Spécification d'accompagnement)	5.5 Article C.8	STSA/(IEC)	10.3.14

10.2 Opérations

10.2.1 Maintenance de certification de produit

La STS Association, en tant qu'Autorité d'enregistrement enregistrée auprès de l'IEC, doit assurer aux utilisateurs de la STS un accès aux services de certification de produit.

Elle doit également assurer que de tels fournisseurs de service sont dûment accrédités et autorisés à fournir ce service et qu'ils se conforment aux exigences de la présente Norme et à tout(e) autre COP ou spécification pertinent(e).

10.2.2 Maintenance du DSN

Le constructeur de compteur à paiement a le contrôle total de sa plage allouée de valeurs de DSN (dans son domaine de MfrCode alloué), il ne nécessite pas d'autre maintenance.

10.2.3 Maintenance du RO

L'entreprise de distribution doit gérer l'usage opérationnel de cet élément de données conjointement avec la date de référence STS.

10.2.4 Maintenance du TI

L'entreprise de distribution doit gérer l'usage opérationnel de cet élément.

10.2.5 Maintenance du TID

L'entreprise de distribution doit gérer l'usage opérationnel de cet élément de données par une programmation appropriée des systèmes de vente de jeton ou des systèmes POS.

10.2.6 Maintenance du SpecialReservedTokenIdentifier

L'entreprise de distribution doit gérer l'usage opérationnel de cet élément de données par une programmation appropriée des systèmes de vente de jeton ou des systèmes POS.

10.2.7 Maintenance du MfrCode

La STS Association, en tant qu'Autorité d'enregistrement enregistrée auprès de l'IEC, doit fournir le service permettant d'allouer les valeurs de MfrCode aux constructeurs de compteurs à paiement et de mettre la liste des valeurs allouées de MfrCode à la disposition des utilisateurs de la STS, sur demande.

10.2.8 Maintenance des tables de substitution

La STS Association, en tant qu'Autorité d'enregistrement enregistrée auprès de l'IEC, doit fournir le service permettant de mettre les valeurs réelles pour le Tableau 33 et le Tableau 41 à la disposition des utilisateurs de la STS, sur demande.

10.2.9 Maintenance des tables de permutation

La STS Association, en tant qu'Autorité d'enregistrement enregistrée auprès de l'IEC, doit fournir le service permettant de mettre les valeurs réelles pour le Tableau 34 et le Tableau 40 à la disposition des utilisateurs de la STS, sur demande.

10.2.10 Maintenance du SGC

La STS Association, en tant qu'Autorité d'enregistrement enregistrée auprès de l'IEC, doit assurer aux utilisateurs de la STS un accès aux services d'allocation de SGC et garantir que les valeurs de SGC sont uniques au niveau global. Ces services sont typiquement fournis par un KMC.

10.2.11 Maintenance de la VendingKey

La STS Association, en tant qu'Autorité d'enregistrement enregistrée auprès de l'IEC, doit assurer aux utilisateurs de la STS un accès aux services d'allocation de VendingKey et garantir que les valeurs de VendingKey sont uniques au niveau global et que les valeurs de VendingKey sont disponibles entre fournisseurs de service de KMC. Ces services sont typiquement fournis par un KMC.

La STS Association doit également assurer la conformité de tels fournisseurs de service aux exigences et recommandations données dans la présente Norme internationale et à tout(e) autre COP ou spécification pertinent(e).

10.2.12 Maintenance du KRN

Cet élément est intrinsèquement couplé à la VendingKey et est géré par le fournisseur de service de KMC, en étant assujetti aux mêmes conditions que dans le cas de la maintenance de la VendingKey.

10.2.13 Maintenance du KT

Cet élément est intrinsèquement couplé à la VendingKey et est géré par le fournisseur de service de KMC, en étant assujetti aux mêmes conditions que dans le cas de la maintenance de la VendingKey.

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC doit administrer tout ajout à la plage des valeurs de KeyType données dans le Tableau 30.

Le processus doit suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par ces organisations.

Une définition de KeyType supplémentaire doit exiger une révision ou un amendement de la présente Norme.

10.2.14 Maintenance du KEN

Cet élément est intrinsèquement couplé à la VendingKey et est géré par le fournisseur de service de KMC, en étant assujetti aux mêmes conditions que dans le cas de la maintenance de la VendingKey.

10.2.15 Maintenance de la KEK

Le fournisseur de service de KMC a exclusivement le contrôle de cet élément de données, car il constitue une partie intégrante de ses opérations de gestion de clé.

La STS Association, en tant qu'Autorité d'enregistrement enregistrée auprès de l'IEC, doit assurer que les fournisseurs de service de KMC se conforment aux exigences de la présente Norme et à tout autre COP pertinent.

10.2.16 Maintenance du CC

La STS Association, en tant qu'Autorité d'enregistrement enregistrée auprès de l'IEC, doit assurer aux utilisateurs de la STS un accès aux services d'allocation de CC et garantir que les valeurs de CC sont uniques au niveau global. Ces services sont typiquement fournis par un KMC.

10.2.17 Maintenance de l'UC

La STS Association, en tant qu'Autorité d'enregistrement enregistrée auprès de l'IEC, doit assurer aux utilisateurs de la STS un accès aux services d'allocation de l'UC et garantir que les valeurs de l'UC sont uniques au niveau global. Un KMC fournit typiquement ces services.

10.2.18 Maintenance du KMCID

La STS Association, en tant qu'Autorité d'enregistrement enregistrée auprès de l'IEC, doit assurer aux utilisateurs de la STS un accès aux services d'allocation du KMCID et garantir que les valeurs du KMCID sont uniques au niveau global. La STS Association fournit typiquement ces services.

10.2.19 Maintenance du CMID

Le constructeur du CM a le contrôle total de l'allocation des valeurs de CMID à ses dispositifs CM manufacturés et il n'y a aucun service en place pour assurer l'unicité de cet élément de données.

Une fois qu'un CM particulier est enregistré dans un système STS (typiquement auprès d'un fournisseur de service de KMC), le CMID est simplement enregistré pour des besoins de référence et aucun service de maintenance supplémentaire sur cet élément de données n'est requis.

10.2.20 Maintenance du CMAC

Le constructeur du CM a le contrôle total de l'allocation des valeurs de CMAC à ses dispositifs CM manufacturés et il n'y a aucun service en place pour assurer l'unicité de cet élément de données.

La transaction d'enregistrement d'une valeur de CMAC est typiquement conduite entre le constructeur du CM et le fournisseur de service de KMC, et elle reste dans le domaine opérations des deux parties prenantes.

La STS Association, en tant qu'Autorité d'enregistrement enregistrée auprès de l'IEC, doit assurer la conformité de ces constructeurs et fournisseurs de services aux exigences et recommandations données dans la présente Norme et tout autre COP pertinent.

10.3 Normalisation

10.3.1 Maintenance de l'IIN

La présente Norme définit une valeur constante pour les compteurs à paiement d'électricité au niveau mondial.

L'ISO peut émettre des valeurs différentes pour d'autres services à la suite d'une demande par les fournisseurs de services.

Tout changement apporté aux règles définies dans la présente Norme exige une révision ou un amendement de la présente Norme.

10.3.2 Maintenance du TCT

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC doit administrer tout ajout à la plage des valeurs de TCT données dans le Tableau 5.

Le processus doit suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par ces organisations.

Une entrée supplémentaire dans le Tableau 5 doit exiger une révision ou un amendement de la présente Norme et une nouvelle partie à la série IEC 62055-5x.

10.3.3 Maintenance du DKGA

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC doit administrer tout ajout à la plage des valeurs de DKGA données dans le Tableau 6.

Le processus doit suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par ces organisations.

Une entrée supplémentaire dans le Tableau 6 doit exiger une révision ou un amendement de la présente Norme.

10.3.4 Maintenance de l'EA

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC doit administrer tout ajout à la plage des valeurs de l'EA données dans le Tableau 7.

Le processus doit suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par ces organisations.

Une entrée supplémentaire dans le Tableau 7 doit exiger une révision ou un amendement de la présente Norme.

10.3.5 Maintenance de la TokenClass

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC doit administrer tout ajout à la plage des valeurs de TokenClass données dans le Tableau 13 et dans le Tableau 14.

Le processus doit suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par ces organisations.

Une définition de TokenClass supplémentaire doit exiger une révision ou un amendement de la présente Norme.

10.3.6 Maintenance de la TokenSubClass

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC doit administrer tout ajout à la plage des valeurs de TokenSubClass données dans le Tableau 14.

Le processus doit suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par ces organisations.

Une définition de TokenSubClass supplémentaire doit exiger une révision ou un amendement de la présente Norme.

10.3.7 Maintenance de l'InitiateMeterTest/DisplayControlField

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC doit administrer tout ajout à la plage des valeurs de l'InitiateMeterTest/DisplayControlField données dans le Tableau 22.

Le processus doit suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par ces organisations.

Une valeur supplémentaire de l'InitiateMeterTest/DisplayControlField doit exiger une révision ou un amendement de la présente Norme.

10.3.8 Maintenance de RegisterToClear

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC doit administrer tout ajout à la plage des valeurs de RegisterToClear données dans le Tableau 23.

Le processus doit suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par ces organisations.

Une valeur supplémentaire de RegisterToClear doit exiger une révision ou un amendement de la présente Norme.

10.3.9 Maintenance de la date de référence STS (STS base date)

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC doit administrer tout changement apporté à la date de référence STS.

Le processus doit suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par ces organisations.

Un changement apporté à la valeur de la date de référence STS doit exiger une révision ou un amendement de la présente Norme.

10.3.10 Maintenance du Rate

Cet élément de données est actuellement réservé pour une définition future.

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC doit administrer tout changement apporté à la définition de l'élément de données Rate.

Le processus doit suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par ces organisations.

Un changement apporté à la définition de l'élément de données Rate doit exiger une révision ou un amendement de la présente Norme.

10.3.11 Maintenance du WMFactor

Cet élément de données est actuellement réservé pour une définition future.

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC doit administrer tout changement apporté à la définition de l'élément de données WMFactor.

Le processus doit suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par ces organisations.

Un changement apporté à la définition de l'élément de données WMFactor doit exiger une révision ou un amendement de la présente Norme.

10.3.12 Maintenance du MFO

Les définitions des instances de l'objet MFO ne relèvent actuellement pas du domaine normatif de la présente Norme et sont mentionnées purement à titre informatif.

La STS Association administre de manière exclusive la définition des instances de MFO selon ses propres procédures normalisées internes relatives à la soumission de propositions d'études nouvelles.

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC peut, à l'avenir, proposer ces instances de MFO à l'IEC en vue de leur développement en Normes internationales, qui doivent suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par l'IEC.

10.3.13 Maintenance du FOIN

L'allocation et l'assignation des valeurs de FOIN ne relèvent actuellement pas du domaine normatif de la présente Norme et sont mentionnées purement à titre informatif.

La STS Association administre de manière exclusive l'allocation et l'assignation des valeurs de FOIN conjointement avec l'enregistrement des instances de MFO comme spécifications d'accompagnement.

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC peut, à l'avenir, proposer ces valeurs de FOIN à l'IEC en vue de leur développement en Normes internationales, qui doivent suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par l'IEC.

10.3.14 Maintenance de la Spécification d'accompagnement

Le développement de spécifications d'accompagnement ne relève actuellement pas du domaine normatif de la présente Norme et est mentionné purement à titre informatif.

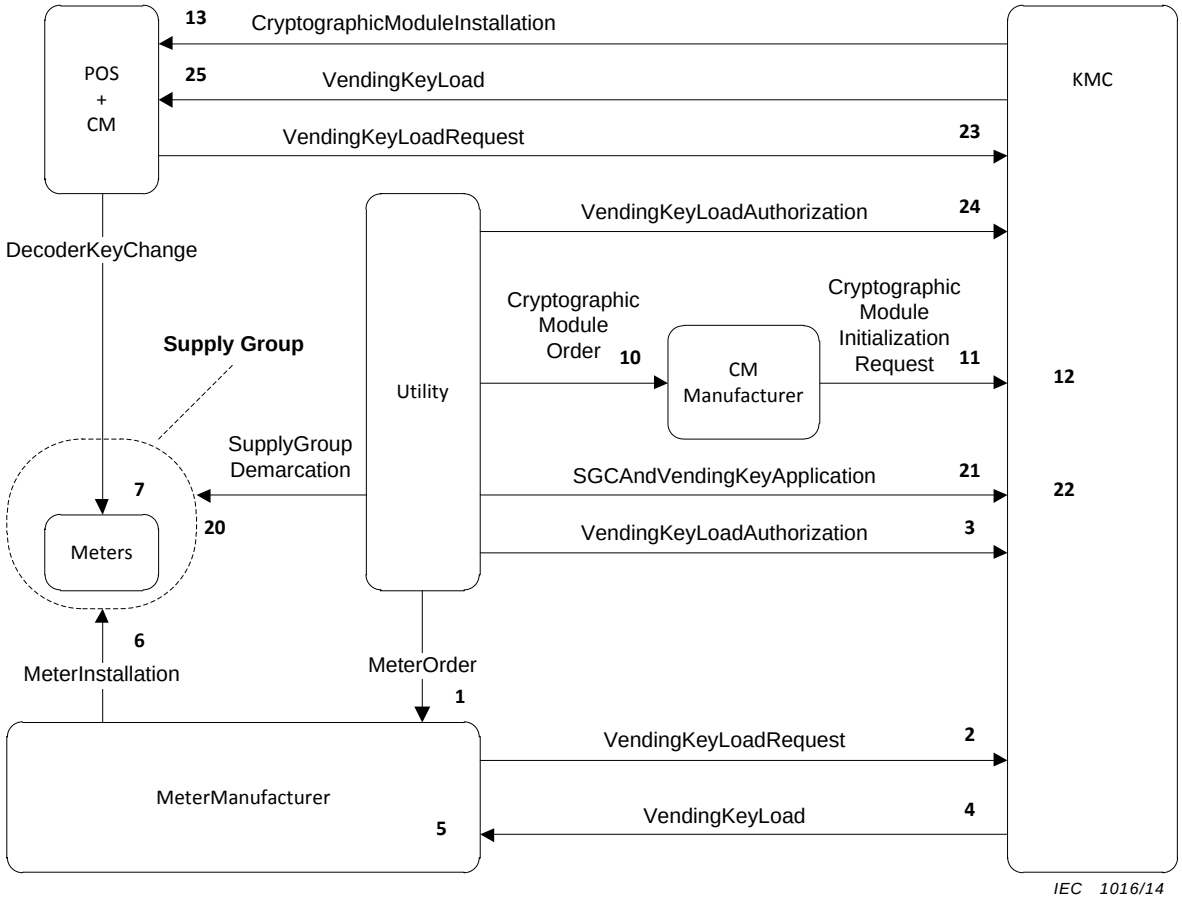
La STS Association administre de manière exclusive le développement de spécifications d'accompagnement conjointement avec l'enregistrement des instances de MFO et l'assignation des valeurs de FOIN.

La STS Association en liaison partenariale avec le groupe de travail WG 15 du CE 13 de l'IEC peut, à l'avenir, proposer ces spécifications d'accompagnement à l'IEC en vue de leur développement en Normes internationales, qui doivent suivre les procédures normalisées relatives à la soumission de propositions d'études nouvelles telles qu'elles sont instituées par l'IEC.

Annexe A
(informative)

Lignes directrices pour un KeyManagementSystem (KMS)

Un diagramme de relations et d'interactions entre des entités est montré à la Figure A.1.



Légende

Anglais	Français
POS+CM	Point de vente + Module cryptographique
CryptographicModuleInstallation	Installation de module cryptographique
VendingKeyLoad	Charge de clé de vente
VendingKeyLoadRequest	Demande de charge de clé de vente
KMC	Centre de gestion de clé
VendingKeyLoadAuthorization	Autorisation de charge de clé de vente
DecoderKeyChange	Changement de clé de décodeur
Cryptographic Module Order	Ordre d'achat de module cryptographique
CM Manufacturer	Constructeur du CM
Cryptographic Module Initialization Request	Demande d'initialisation du module cryptographique
Supply Group	**Groupe d'approvisionnement**
SupplyGroup Demarcation	Délimitation du groupe d'approvisionnement
Utility	Entreprise de distribution
SGCAndVendingKeyApplication	Application de clé de vente et de SGC

Anglais	Français
Meters	Compteurs
MeterInstallation	Installation de compteur
MeterOrder	Ordre d'achat de compteur
MeterManufacturer	Constructeur du compteur

Figure A.1 – KeyManagementSystem et relations interactives entres des entités

Les entités qui jouent un rôle dans le processus de KMS sont données dans le Tableau A.1.

Tableau A.1 – Entités qui participent aux processus de KMS

Entité	Rôle / Nom
Utility (c'est-à-dire: Entreprise de distribution)	Fournisseur d'un service tel que l'électricité
MeterManufacturer	Constructeur de compteurs à paiement / dispositifs décodeurs
CMManufacturer	Constructeur de modules cryptographiques
KMC	KeyManagementCentre (Centre de gestion de clé)
CM	CryptographicModule (Module cryptographique)
POS	PointOfSale (Point de vente)
Meter (c'est-à-dire: Compteur)	Compteur à paiement

Les processus de compteur à paiement et les processus de DecoderKey sont donnés dans le Tableau A.2.

Tableau A.2 – Processus entourant le compteur à paiement et la DecoderKey

Numéro de processus	Contexte
1	MeterOrder L'entreprise de distribution passe un ordre d'achat de compteurs à paiement auprès du MeterManufacturer. L'ordre d'achat stipule que les compteurs à paiement sont chargés avec des valeurs de DDTK, DUTK ou DCTK pour le SGC spécifié.
2	VendingKeyLoadRequest Le MeterManufacturer demande la VendingKey (VUDK ou VCDK) pour le SGC spécifique, si requise, auprès du KMC; autrement, il utilise sa propre VDDK allouée (voir 6.5.2.2)
3	VendingKeyLoadAuthorization L'Utility (c'est-à-dire: entreprise de distribution) autorise le KMC à charger les valeurs demandées de VendingKey et à les descendre au MeterManufacturer
4	VendingKeyLoad Les valeurs demandées de VendingKey sont chargées dans l'équipement de fabrication sécurisé, certifié STS du MeterManufacturer
5	DecoderKeyLoad Le MeterManufacturer génère les valeurs de DDTK, DUTK ou DCTK à partir des valeurs de VDDK, VUDK ou VCDK conformément à l'ordre d'achat du compteur à paiement et les charge dans le compteur à paiement (voir 6.5.3)
6	MeterInstallation Les compteurs à paiement sont livrés à l'Utility et installés dans le SupplyGroup délimité
7	DecoderKeyChange Si c'est requis, la valeur de DecoderKey peut être changée par les KeyChangeTokens de vente issus de l'équipement du POS (voir en 6.2.7 et 6.2.8 les jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey). Voir aussi les processus 23 à 25 ci-dessous relatifs au chargement de la VendingKey

Les processus du CryptographicModule sont donnés dans le Tableau A.3.

Tableau A.3 – Processus entourant le CryptographicModule

Numéro de processus	Contexte
10	CryptographicModuleOrder L'Utility (ou le constructeur de POS) passe un ordre d'achat de module cryptographique auprès d'un constructeur de module cryptographique
11	CryptographicModuleInitialisationRequest Le CryptographicModule est envoyé au KMC pour être initialisé avec les valeurs de clé secrète, qui sont utilisées par la suite pour distribuer en toute sécurité les valeurs de VendingKey du KMC vers le CryptographicModule
12	CryptographicModuleAuthenticationAndInitialization Le KMC vérifie que le CryptographicModule est authentique et l'initialise alors avec des valeurs de clé secrète, qui sont utilisées par la suite pour distribuer en toute sécurité les valeurs de VendingKey du KMC vers le CryptographicModule (voir la KEK dans l'Annexe B)
13	CryptographicModuleInstallation Le CryptographicModule est installé et est prêt pour charger les valeurs de VendingKey issues du KMC en utilisant typiquement les KeyLoadFiles (voir KLF dans l'Annexe B)

Les processus du SGC de la VendingKey sont donnés dans le Tableau A.4.

Tableau A.4 – Processus entourant le SGC et la VendingKey

Numéro de processus	Contexte
20	SupplyGroupDemarcation L'Utility alimente en électricité un groupe défini de ses consommateurs. Elle décide de la taille et des frontières du groupe selon des considérations de risque à la sécurité et de protection de recettes, l'emplacement géographique et les caractéristiques logistiques du réseau
21	SGCAndVendingKeyApplication L'Utility fait une demande au KMC d'un SGC de type spécifié (unique ou commun) et d'une VendingKey associée d'un type spécifié (VUDK ou VCDK; voir 6.5.3)
22	SGCAndVendingKeyAllocation Le KMC alloue un SGC et une VendingKey secrète associée du KT requis au demandeur et stocke les éléments dans ses enregistrements
23	VendingKeyLoadRequest L'opérateur de POS demande la valeur de VendingKey (VDDK, VUDK ou VCDK) pour le SGC particulier auprès du KMC qui lui permet de vendre à des compteurs à paiement chargés avec la valeur de DecoderKey associée (DDTK, DUTK ou DCTK)
24	VendingKeyLoadAuthorization L'Utility autorise le KMC à charger les valeurs de VendingKey demandées (VUDK ou VCDK). En variante, le MeterManufacturer autorise le KMC à charger la valeur de VDDK demandée.
25	VendingKeyLoad Les valeurs de VendingKey demandées sont chargées dans le CryptographicModule qui est utilisé par l'équipement de POS pour générer des jetons pour les compteurs à paiement dans le SupplyGroup.

Les exigences obligatoires relatives à un KeyManagementSystem sont spécifiées à l'Article 9.

Voir aussi l'Article C.2 Code de bonnes pratiques pour plus d'informations concernant la gestion des VendingKey.

Voir aussi C.2.2.1 Code de bonnes pratiques pour plus d'informations concernant les lignes directrices de démarcation de SGC.

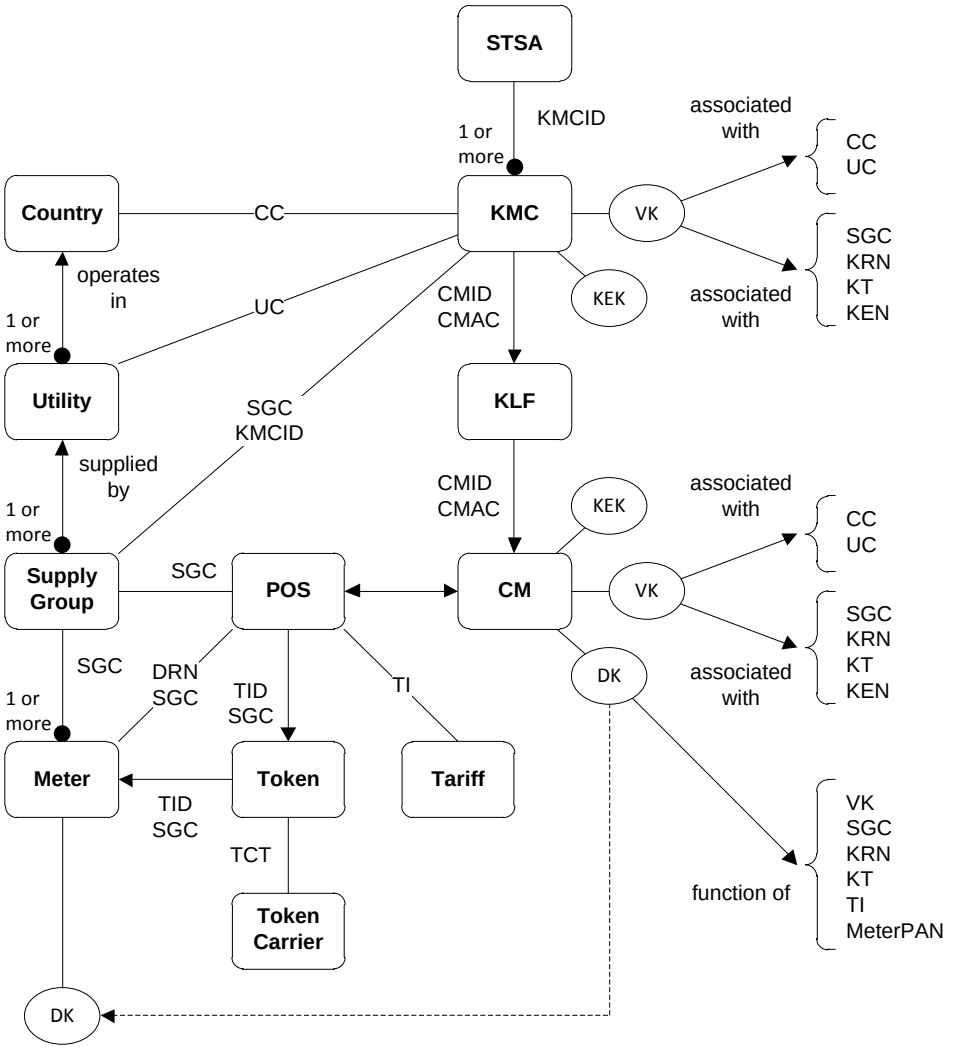
Voir aussi l'Annexe B pour plus d'informations concernant les entités et les identificateurs dans un système conforme à la STS.

Voir aussi l'Article 10 pour la maintenance des entités STS et des services connexes.

Annexe B
(informative)

Entités et identificateurs dans un système conforme à la STS

Les entités et les identificateurs appropriés déployés dans un système conforme à la STS sont montrés à la Figure B.1.



IEC 1017/14

Légende

Anglais	Français
associated with	associé à
function of	fonction de
Country	Pays
Utility	Entreprise de distribution
Supply Group	Groupe d'approvisionnement
Meter	Compteur
supplied by	fourni par

Anglais	Français
1 or more	1 ou plus
operates in	opère dans
Token	Jeton
Tariff	Tarif
Token Carrier	Support de jeton

Figure B.1 – Entités et identificateurs déployés dans un système conforme à la STS

Pour la maintenance de ces entités et de ces services connexes, voir l'Article 10.

Les entités qui sont typiquement déployées dans un système conforme à la STS sont données dans le Tableau B.1.

Tableau B.1 – Entités types déployées dans un système conforme à la STS

Entité	Contexte	Référence
Country (c'est-à-dire: Pays)	Zone géographique avec des frontières politiquement délimitées, qui peut changer au fil du temps	x
Utility (c'est-à-dire: Entreprise de distribution)	Entité qui fournit un service comme l'électricité à son consommateur final au moyen d'un compteur à paiement. Une ou plusieurs entreprises de distribution sont opérationnelles dans un pays. Les entreprises de distribution changent leurs identités constitutionnelles au fil du temps	x
SupplyGroup (Groupe d'alimentation)	Un sous-groupe de compteurs à paiement au sein d'un réseau de distribution. Une Utility peut effectuer une fourniture à un ou plusieurs SupplyGroup. Un SupplyGroup peut changer sa relation à un Country et à une Utility au fil du temps.	6.1.6
Meter (c'est-à-dire: Compteur)	Le compteur à paiement utilisé pour commander la livraison ou l'approvisionnement du service au consommateur final (voir aussi l'IEC 62055-31). Un ou plusieurs compteurs à paiement sont groupés dans un SupplyGroup. Un compteur à paiement peut passer à un SupplyGroup différent au moyen d'un changement de DecoderKey correspondante.	IEC 62055-31 IEC 62055-21
POS	Dispositif de PointOfSale qui est capable de générer des jetons pour n'importe quel compteur à paiement dans un SupplyGroup, en ayant accès à la valeur de VendingKey pour le SupplyGroup particulier. Il est techniquement et pratiquement faisable qu'un POS puisse avoir accès aux valeurs de VendingKey de plus d'un SupplyGroup et être ainsi capable de générer également des jetons pour des compteurs à paiement appartenant à ces SupplyGroup. Les VendingKey peuvent passer d'un dispositif de PointOfSale à un autre au fil du temps, en fonction de la relation commerciale entre un fournisseur et une Utility particulière.	IEC 62055-21
TokenCarrier	Le dispositif physique, ou support sur lequel sont codées les informations relatives au jeton et qui est ensuite utilisé pour transférer le jeton vers le compteur à paiement. Cela peut se présenter sous la forme d'une chaîne numérique imprimée ou d'une carte à codage magnétique qui est transportée à la main vers le compteur à paiement et insérée manuellement dans le dispositif de lecture du compteur à paiement par l'utilisateur (consommateur final), ou il peut s'agir d'un support de jeton virtuel sous la forme d'une connexion de communication directe à un dispositif de client distant.	3.1
Token	Jeton tel que défini dans la présente Norme au moyen duquel le dispositif de POS est capable de transférer des instructions et des informations au compteur à paiement ou récupérer des informations du compteur à paiement	3.1

Entité	Contexte	Référence
Tariff (c'est-à-dire: Tarif)	La formule utilisée pour calculer la charge par unité de service. Dans le cas des compteurs à paiement unidirectionnels, le tarif est normalement appliqué au POS au moment où le consommateur final achète un jeton. Il y a normalement plusieurs structures tarifaires selon les différentes catégories et les différents contrats de consommateurs. Chaque tarif est donc associé à un TI (voir ci-dessous) pour la facilité de la référence.	6.1.7 6.2.6
STSA	Association de spécification de transfert normalisé (Standard Transfer Specification Association) qui tient un registre de tous les KMC, qui sont déployés au niveau global	Article C.1
KMC	KeyManagementCentre. L'infrastructure qui est chargée de gérer et commander le KeyManagementSystem	Article 9 Annexe A Article C.2
KLF	KeyLoadFile. Le mécanisme sécurisé utilisé par le KMC pour distribuer les valeurs de VendingKey aux modules cryptographiques	Annexe A
CM	CryptographicModule. Le dispositif sécurisé utilisé par le POS pour générer des valeurs de DecoderKey à partir de valeurs de VendingKey et générer des jetons à partir des valeurs de DecoderKey	Annexe A
KEK	KeyExchangeKey. Une valeur de double clé DES 64 bits secrète partagée entre le KMC et le CM, qui est utilisée pour chiffrer des valeurs de VendingKey qui sont distribuées au moyen du KLF (voir KLF ci-dessus)	x
VK	VendingKey. Une valeur de clé DES 64 bits, générée, stockée et distribuée par le KMC à d'autres modules cryptographiques dans des conditions maîtrisées et autorisées, le cas échéant. Elle est utilisée pour générer des valeurs de DecoderKey à l'intérieur du CM	6.5.2.2
DK	DecoderKey. Une valeur de clé STA 64 bits ou une valeur de clé DES 64 bits générée en fonction de plusieurs valeurs: DK = f (VK, SGC, KRN, KT, TI, MeterPAN). Elle est partagée entre le CM et le compteur à paiement et elle est utilisée pour chiffrer et déchiffrer les jetons qui sont envoyés du POS vers le compteur à paiement ou du compteur à paiement vers le POS	6.5.2.3

Les identificateurs qui sont associés aux entités ci-dessus sont donnés dans le Tableau B.2.

Tableau B.2 – Identificateurs associés aux entités dans un système conforme à la STS

Identificateur	Contexte	Référence
CC	CountryCode (Code de pays) Un code identifiant de façon unique le pays dans lequel l'Utility opère et où les compteurs à paiement sont installés. Il est enregistré dans le KMC et associé à VK au niveau du KMC et du CM	x
UC	UtilityCode (Code d'entreprise de distribution) Un code alloué par le KMC pour identifier de façon unique l'Utility spécifique à laquelle la VK et le SGC sont alloués. Il est enregistré dans le KMC et est associé à VK au niveau du CM	x
KMCID	KeyManagementCentreIdentifier Identificateur unique pour chaque KMC dans le monde. Chaque KMCID est enregistré auprès de la STSA	x
CMID	CryptographicModuleIdentifier (Identificateur de module cryptographique) Identificateur unique pour chaque module cryptographique dans le système	x
CMAC	CryptographicModuleAuthenticationCode (Code d'authentification de module cryptographique) Un ensemble de codes secrets que le KMC et le POS peuvent utiliser pour authentifier le CM avant de lui confier d'autres valeurs secrètes. Les exemples types sont DeviceAuthenticationCode et FirmwareAuthenticationCode.	x
TID	TokenIdentifier (Identificateur de jeton) Identificateur unique basé sur le temps pour chaque jeton. Il est partagé entre le POS, le jeton et le compteur à paiement.	6.3.5.1
MeterPAN	MeterPrimaryAccountNumber Un numéro d'identification unique pour chaque compteur à paiement conforme à la STS. Il est partagé entre le compteur à paiement et le POS. Le fait de le coder dans la DecoderKey applique l'association avec le compteur à paiement	6.1.2
DRN	DecoderReferenceNumber Le numéro unique tel qu'il apparaît dans le MeterPAN. Il est partagé entre le POS et le compteur à paiement	6.1.2.3
TCT	TokenCarrierType (Type de support de jeton) Le type de support utilisé sur lequel le jeton est codé pour le transfert au compteur à paiement	6.1.3
SGC	SupplyGroupCode (Code de groupe d'alimentation) Nombre unique alloué par le KMC pour identifier un SupplyGroup de l'Utility. Il est partagé entre le SupplyGroup, le KMC et le POS. Il est associé à la valeur de VendingKey et est enregistré dans le KMC et également dans le CM. Le fait de le coder dans la DecoderKey applique l'association avec le compteur à paiement	6.1.6
TI	TariffIndex (Index de tarifs) Le numéro d'index à un registre de tarifs associé à un Tariff particulier pour chaque consommateur. Il est partagé entre le Tariff et le POS. Le fait de le coder dans la DecoderKey applique l'association avec le compteur à paiement. Cela signifie que la DecoderKey doit changer si le client passe à une structure tarifaire différente.	6.1.7
KRN	KeyRevisionNumber (Numéro de révision de clé) Révision de la VendingKey telle qu'allouée par le KMC. Elle est associée à la valeur de VendingKey au niveau du KMC et du CM. Le fait de la coder dans la DecoderKey applique l'association avec le compteur à paiement	6.1.8
KT	KeyType (Type de clé) Le type de la VendingKey tel qu'alloué par le KMC. Il est associé à la valeur de VendingKey au niveau du KMC et du CM. Le fait de le coder dans la DecoderKey applique l'association avec le compteur à paiement	6.1.9

Identificateur	Contexte	Référence
KEN	KeyExpiryNumber Numéro qui est associé à une durée de validité pour la VendingKey. Il est associé à la valeur de VendingKey au niveau du KMC et du CM. Il n'est pas codé dans la DecoderKey, mais il est transféré au DecoderKeyRegister au moyen des jetons Set1stSectionDecoderKey et Set2ndSectionDecoderKey.	6.1.10

Annexe C
(informative)

Code de bonnes pratiques pour la mise en œuvre des systèmes conformes à la STS

C.1 Services de maintenance et d'assistance fournis par la STS Association

La STS Association est une entreprise à but non lucratif constituée en Afrique du Sud dont les membres sont des constructeurs de compteurs à paiement et de systèmes de vente associés et des entreprises de distribution. L'objet de la STS Association est de promouvoir l'utilisation de la STS, développer davantage la fonctionnalité et maintenir l'infrastructure requise pour fournir des services de soutien comme la gestion de clé, la certification de produit et la normalisation aux utilisateurs de la STS.

Voir aussi l'Article 10 pour plus de détails relatifs à la maintenance des entités STS et des services connexes.

Le Secrétariat général peut être contacté à l'adresse indiquée dans l'introduction de la présente norme. Le courriel est le mécanisme préférentiel pour la correspondance avec l'Association.

C.2 Gestion de clé

C.2.1 Services de gestion de clé

(Voir aussi l'Annexe A.)

La STS Association exploite un KMC et fournit les services de gestion de clé aux entreprises de distribution et aux constructeurs de produits conformes à la STS dans le monde entier conformément à la présente Norme.

C.2.2 Distribution de SupplyGroupCode et de VendingKey

C.2.2.1 Éléments de données associés à un SGC

(Voir aussi 6.1.6)

Le KMC assure une allocation unique de valeurs de SGC selon la présente Norme.

Le KMC génère, enregistre et distribue des valeurs de VDDK, de VUDK et de VCDK avec les KRN, KT et KEN associés selon la présente Norme.

Le KMC assure que les valeurs de VendingKey sont disponibles à tous les constructeurs de produits certifiés STS selon la présente Norme.

Afin de gérer efficacement la génération, le stockage et la distribution des valeurs de SGC et de VendingKey, il est recommandé d'enregistrer et d'associer de façon unique à un SGC les éléments de données consignés dans le Tableau C.1.

Tableau C.1 – Éléments de données associés à un SGC

Élément	Contexte	Réf
SGC	Valeur réelle du SupplyGroupCode enregistrée dans le KMC	6.1.6
Country (c'est-à-dire: Pays)	CountryCode comme étant le pays où le SGC et la VendingKey sont à utiliser	Annexe B
Location (c'est-à-dire: Emplacement)	Endroit associé à la délimitation du SupplyGroup (Pays, État, Province, Cité, Ville, Banlieue)	x
Network (c'est-à-dire: Réseau)	Réseau associé à la délimitation du SupplyGroup (nom, ID)	x
Owner (c'est-à-dire: Propriétaire)	À qui ce SGC est alloué: UtilityCode (si applicable) Nom de l'organisation (entreprise de distribution) Adresse (postale, physique, site web) Personne à contacter et coordonnées (nom, adresse postale, courriel, tél, télécopie) Signataire d'autorisation (nom, coordonnées du contact)	x
OwnerHistory (c'est-à-dire: Historique des propriétaires)	Enregistrement de changements apportés à l'association du SGC à des droits de propriété au fil du temps	X
LocationHistory (c'est-à-dire: Historique des emplacements)	Enregistrement de changements apportés à l'association du SGC à des emplacements au fil du temps	X
NetworkHistory (c'est-à-dire: Historique des réseaux)	Enregistrement de changements apportés à l'association du SGC à des réseaux au fil du temps	X
KMC	KMCID et pays d'origine du KMC comme source du SGC et de la VendingKey	Article 9 Annexe A
VendingKey	VendingKey plus attributs (KRN, KT, KEN). Ces valeurs sont en format chiffré	6.5.2 6.1.8 6.1.9 6.1.10
SGCDistributionRegister (c'est-à-dire: Registre de distribution de SGC)	Registre de SGC en fonction d'ID de CM (c'est-à-dire à quels modules cryptographiques un SGC particulier a été distribué au fil du temps)	x

C.2.2.2 Lignes directrices relatives à la délimitation de SupplyGroupCode

Ce sujet est traité de façon complète dans le Code de bonnes pratiques de la STS Association (voir Bibliographie). Dans le souci de donner un certain nombre d'indicateurs ici, un certain nombre de facteurs à prendre en considération sont donnés ci-dessous.

Facteurs à considérer pour décider des délimitations de SGC:

- risque à la sécurité en termes de compromission d'une VendingKey;
- risque à la sécurité en termes de dispositifs POS volés;
- logistique pour les pièces de rechange de compteurs à paiement;
- contrôle des agents de vente de POS pour les autoriser à vendre à un groupe;
- logistique pour séparer les recettes recueillies des agents de vente de POS;
- logique métier particulière autour de la logistique de maintenance et d'approvisionnement de réseau de distribution;

- règles de vente croisée relatives aux frontières de SGC;
- changement de droits de propriété du compteur à paiement au fil du temps (marchés dérégulés),
- changement de fournisseur au fil du temps (marchés dérégulés).

C.2.3 Distribution de CryptographicModule

(Voir aussi l'Annexe A.)

Afin de gérer efficacement la distribution des valeurs de SGC et de VendingKey à des modules cryptographiques, il est recommandé d'enregistrer les éléments de données consignés dans le Tableau C.2.

Tableau C.2 – Éléments de données associés au CryptographicModule

Élément	Contexte	Référence
CM	Attributs du CryptographicModule (CMID, CMType, HardwareVersion, Softwareversion, KEK, FAC, DAC).	Annexe A Annexe B
CMManufacturer	Nom et coordonnées de contact de l'organisation	Annexe A
CMOwner	À qui ce CM appartient: UtilityCode (si applicable) Nom de l'organisation (entreprise de distribution) Adresse (postale, physique, site web) Personne à contacter et coordonnées (nom, adresse postale, courriel, tél, télécopie) Personne responsable (nom, coordonnées du contact)	Annexe A
CMLocation	Coordonnées de la destination prévue du CM où il sera utilisé (pays, état, province, cité, ville, banlieue)	X
KMC	KMCID et pays d'origine qui ont initialisé le CM particulier	Article 9 Annexe A
CMOwnerHistory	Registre historique des changements de droits de propriété apportés aux modules cryptographiques au fil du temps	x
CMLocationHistory	Registre historique des changements d'emplacements apportés aux modules cryptographiques au fil du temps	x

C.2.4 Expiration de clé

(Voir aussi 6.1.10, 6.5.2.6, 7.3.1.1).

Au cas où l'expiration de clé pour les VendingKey n'est pas mise en œuvre de façon dynamique dans une installation conforme à la STS, la pratique recommandée est de mettre le KEN à 255.

À la date de la publication de la présente Norme, l'option d'expiration de clé pour les DecoderKey dans les compteurs à paiement n'avait été mise en œuvre dans aucune installation conforme à la STS.

C.3 MeterPAN

C.3.1 Pratique générale

(Voir aussi 6.1.2).

Le MeterPAN sert à identifier de façon unique chaque compteur à paiement dans l'installation conforme à la STS dans le monde entier, en étant ainsi capable d'étiqueter et acheminer les transactions en conséquence. Tous les utilisateurs de la STS sont ainsi encouragés à suivre cette pratique, qui va dans le sens de celle de la gestion de transactions bancaires et financières (voir également l'ISO 4909).

C.3.2 IssuerIdentificationNumbers

Comme cela a été clarifié en 6.1.2.2, l'IIN pour les codes de constructeur de 2 chiffres doit être 600727. Pour les codes de constructeur de 4 chiffres, l'IIN doit être 0000.

C.3.3 ManufacturerCodes

(Voir aussi 6.1.2.3.2).

Les valeurs de MfrCode sont allouées et gérées par la STS Association pour assurer l'unicité de la série au niveau global et, donc, assurer l'unicité du MeterPAN au niveau global. Noter que les codes de constructeur de 2 chiffres et de 4 chiffres peuvent exister tous les deux.

La liste actuelle des valeurs de MfrCode peut être visualisée sur le site Web de la STS ou être obtenue de la STS Association par l'une des voies de contact énumérées ci-dessus.

C.3.4 DecoderSerialNumbers

(Voir aussi 6.1.2.3.3).

Chaque MeterManufacturer gère une plage de nombres de huit chiffres à sa discrétion, tant qu'elle est conforme aux exigences de la présente Norme.

C.4 SpecialReservedTokenIdentifier

(Voir aussi 6.3.5.2).

Chaque entreprise de distribution est libre de déterminer les règles relatives à la façon dont ce SpecialReservedTokenIdentifier est à utiliser comme une application spéciale pour satisfaire à ses besoins spéciaux.

Un exemple d'utilisation de ce SpecialReservedTokenIdentifier dans une application spéciale est comme suit: chaque ménage dans une installation peut bénéficier d'une subvention gouvernementale sous la forme d'un jeton gratuit de la valeur de 50 KWh par mois. Un tel jeton peut être recueilli à n'importe quel jour du mois et autant de fois que souhaité, mais il convient que le compteur à paiement n'accepte que le premier jeton de ce type chaque mois. Une solution à ce problème est d'imposer la règle selon laquelle le SpecialReservedTokenIdentifier est à utiliser pour ce type de jeton dans cette installation particulière. Un tel jeton peut alors être généré à tout moment du mois, parce qu'il utilise toujours le marqueur temporel du 1er jour 00h01 et le compteur à paiement accepte seulement le premier jeton ainsi généré et rejette toutes les copies ultérieures comme étant "Used" ("utilisées").

C.5 Tables de permutation et de substitution pour le STA

La STS Association est enregistrée auprès de l'IEC comme une Autorité d'enregistrement pour fournir des services de maintenance à l'appui des séries IEC 62055-4x et IEC 62055-5x. Comme partie intégrante de ce service, la STS Association fournit aux utilisateurs de la norme, sur demande, les valeurs réelles pour les tables de permutation et de substitution (Tableau 33, Tableau 34, Tableau 40 et Tableau 41) requises en 6.5.4.2, 6.5.4.3, 7.3.3.2 et

7.3.3.3. Les coordonnées de contact pour la STS Association sont données à l'Article C.1 ou peuvent être obtenues sur le site web de l'IEC.

C.6 Codes EA

(Voir aussi 6.1.5).

À mesure que la présente Norme évolue, les codes EA requis sont plus nombreux. Il convient que cela se produise par la voie normale, passant par les Comités nationaux, vers le CE 13 de l'IEC comme propositions d'études nouvelles.

C.7 Codes de TokenCarrierType

(Voir aussi 6.1.3).

À mesure que la présente Norme évolue, les codes TCT requis sont plus nombreux. Il convient que cela se produise par la voie normale, passant par les Comités nationaux, vers le CE 13 de l'IEC comme propositions d'études nouvelles.

C.8 Instances de MeterFunctionObject / spécifications d'accompagnement

Un MeterFunctionObject (MFO) est une spécification orientée objet qui encapsule une certaine fonctionnalité d'un compteur à paiement. Chaque MFO est défini dans une spécification d'accompagnement et reçoit un FunctionObjectIdentificationNumber (FOIN) unique qui lui est alloué.

La STS Association administre l'enregistrement des instances de MFO et réserve les droits exclusifs d'allouer des valeurs de FOIN sous la forme de spécifications d'accompagnement.

Une instance de MFO est proposée à la STS Association comme NWIP, après quoi elle reçoit un FOIN unique qui lui est assigné. La STS Association édite alors le MFO sous la forme d'une spécification d'accompagnement.

Voir aussi la STS 200-1 (voir Bibliographie) pour plus d'informations concernant les classes d'objet fonction et la STS 201-15.1.0 (voir Bibliographie) pour un exemple de spécification d'accompagnement.

C.9 TariffIndex

(Voir aussi 6.1.7).

L'entreprise de distribution a le choix de 2 options:

- relier le TI à sa liste de structures tarifaires et ainsi relier chaque consommateur à un TI. Cela signifie que la DecoderKey doit changer si le consommateur passe d'une structure tarifaire à une autre, parce que le TI associé changera;
- fixer le TI à une valeur constante, disons, = 01 pendant la durée de vie de l'installation du compteur à paiement et ensuite relier chaque consommateur à la liste de structures tarifaires dans le système de gestion, indépendamment du TI. Cela signifie qu'il ne faut pas que la DecoderKey change lors du passage d'un consommateur d'une structure tarifaire à l'autre.

À la date de la publication de la présente Norme, la plupart des entreprises de distribution préfèrent suivre l'option 2. La principale considération est que c'est une opération logistique

majeure que d'effectuer un changement de clé sur un compteur à paiement qui est déjà installé et c'est pourquoi cela tend à être évité dans la mesure du possible.

C.10 Certification de conformité à la STS

C.10.1 Services de certification IEC

L'IEC ne fournit pas de services de certification pour des produits en tant que tels et se repose donc sur des moyens extérieurs pour le faire.

C.10.2 Produits

La STS Association fournit le service aux constructeurs des produits pour faciliter les essais et fournit la certification STS sur la base des résultats d'essai.

C.10.3 Autorité de certification

En temps voulu, la STS Association est en mesure d'habiliter les agents qui peuvent fournir des services de certification STS en son nom.

C.11 Options d'approvisionnement pour les utilisateurs de systèmes conformes à la STS

La présente Norme permet la prise en charge d'une diversité d'options, dont le détail a besoin d'être spécifié au moment où les produits et les systèmes sont achetés auprès des constructeurs et des fournisseurs.

Comme guide général pour les ordres d'achat ou les spécifications d'offres, les éléments donnés dans le Tableau C.3 sont notés.

Tableau C.3 – Éléments qu'il convient de noter dans les ordres d'achat et les soumissions d'offres

Élément	Contexte	Référence
EA	Quel algorithme est à utiliser pour le chiffrement de jeton dans le système de vente et pour le déchiffrement dans le compteur à paiement. Options: • Code STA 07; • Code DEA 09. Il convient que l'acheteur s'assure qu'une exigence de la spécification de soumission d'offre pour les compteurs est que l'étiquetage du compteur à paiement doit inclure le code EA approprié	6.1.5
TCT	Quel TokenCarrierType il convient que le compteur à paiement ou le système de vente prenne en charge. Options: • type de carte magnétique 01; • type numérique 02	6.1.3
DKGA	Quel algorithme il convient que le MeterManufacturer ou le système de vente utilise pour générer la DecoderKey; Options: • DEA (DKGA01); seulement pour les systèmes de vente desservant des compteurs à paiement hérités; • DEA (DKGA02); systèmes actuels; • TDEA (DKGA03);systèmes futurs	6.1.4

Élément	Contexte	Référence
CC	CodeCountry de destination auquel le SGC est à associer au niveau du KMC. Options: • l'un du jeu normalisé de codes pays ISO	Annexe B
UC	UtilityCode auquel le SGC est à associer au niveau du KMC. Options: • UC existant alloué par le KMC; • nouvel UC alloué par le KMC	Annexe B
KMCID	Quel KMC est à utiliser pour obtenir la VendingKey et le SGC. Le MeterManufacturer et le système de vente ont besoin de la VendingKey spécifique pour générer des DecoderKey. Options: • 001; KMC sud-africain actuellement en exploitation; • xxx; KMC de choix ou de pertinence possible dans le futur	Annexe B
SGC	Quel SGC convient-il que le MeterManufacturer ou le système de vente utilise pour générer les DecoderKey? Options: • xxxxxx SGC existant; obtenu du KMC; • nouveau SGC; pour les nouveaux projets, demander à KMC. Quel KT est associé ou convient-il d'associer à ce SGC? Options: • "default" (c'est-à-dire: par défaut); clé du MeterManufacturer; • "unique"; clé d'entreprise de distribution; • "common" (c'est-à-dire: commun); clé d'entreprise de distribution	6.1.6
TI	Quel TariffIndex le MeterManufacturer et le système de vente sont-ils tenus d'utiliser pour générer des DecoderKey? Options: • 00-99; (nouveau); • 00-99; (existant); • relier le TI au tableau tarifaire dans le système de vente; (NOTE 1); • ne pas relier le TI au tableau tarifaire dans le système de vente. (NOTE 2). NOTE 1 Lorsque le TI est relié au tableau tarifaire dans la base de données du système de vente, le consommateur ne peut être déplacé vers une structure tarifaire différente que par allocation d'un autre TI associé. Cela signifie que la DecoderKey a besoin d'être changée en conséquence. NOTE 2 Lorsque le TI n'est pas relié au tableau tarifaire dans la base de données du système de vente, le consommateur peut être déplacé vers une structure tarifaire différente sans être alloué à un autre TI associé. Cela signifie que la DecoderKey peut ne pas être changée.	6.1.7
KRN	Quel KeyRevisionNumber le MeterManufacturer et le système de vente sont-ils tenus d'utiliser pour générer des DecoderKey? Ces informations sont associées à la VendingKey du SGC et elles sont sous le contrôle du KMC auprès duquel il convient de les obtenir	6.1.8
KT	Quel KT le MeterManufacturer et le système de vente sont-ils tenus d'utiliser pour générer des DecoderKey? Ces informations sont associées à la VendingKey du SGC et elles sont sous le contrôle du KMC auprès duquel il convient de les obtenir	6.1.9
KEN	Quel KeyExpiryNumber le MeterManufacturer et le système de vente sont-ils tenus d'utiliser pour générer des DecoderKey? Ces informations sont associées à la VendingKey du SGC et elles sont sous le contrôle du KMC auprès duquel il convient de les obtenir	6.1.10

Élément	Contexte	Référence
DecoderKey expiry (c'est-à-dire: Expiration de DecoderKey)	S'il convient que les DecoderKey expirent ou non. Options: • ne doivent pas expirer (c'est la pratique recommandée actuellement); • doivent expirer (cela implique des changements périodiques de DecoderKey)	6.1.10
VendingKey expiry (c'est-à-dire: Expiration de VendingKey)	S'il convient que les VendingKey expirent ou non. Options: • ne doivent pas expirer (c'est la pratique recommandée actuellement); • doivent expirer (ce cas n'est pas pris en charge actuellement)	6.1.10
Meter dispatching key (c'est-à-dire: clé d'expédition de compteur)	Quel type de DecoderKey il convient que le MeterManufacturer charge dans le compteur à paiement. Options: • DDTK (clé par défaut du constructeur); • DUTK (clé "Unique" d'entreprise de distribution); • DCTK (clé "Common" d'entreprise de distribution)	6.1.6
Tokens (Jetons)	Quels Token (jetons) il convient que le compteur à paiement ou le système de vente prenne en charge. Options: • TransferCredit; • InitiateMeterTest/Display; • SetMaximumPowerLimit; (facultatif) • ClearCredit; • SetTariffRate; (seulement les compteurs à paiement de comptabilisation à base de monnaie) • Set1stSectionDecoderKey; • Set2ndSectionDecoderKey; • ClearTamperCondition; (facultatif) • SetMaximumPhasePowerUnbalanceLimit; (facultatif pour les phases multiples) • SetWaterMeterFactor. (seulement les compteurs à paiement d'eau)	6.2.1
Vending classification (c'est-à-dire: Classification de vente)	Quelles fonctions il convient que les systèmes de vente prennent en charge. Options: • vente; (vente de jetons de crédit) (signalée par “V”); • ingénierie; (vente de jetons de gestion) (signalée par “E”); • changement de clé. (vente de jetons de changement de clé) (signalée par “K”); Un système de vente conforme à la STS peut fournir n'importe quelle combinaison d'une ou toutes les options énumérées. En cas d'approbation par la STS Association, les lettres correspondantes peuvent être affichées sur le logo STS	x
Credit transfer (c'est-à-dire: Transfert de crédit)	Quels types de jetons de TransferCredit il convient que les compteurs à paiement ou le système de vente prennent en charge. Options: • électricité; • eau; • gaz; • temps; • monnaie	6.2.2
Test/display options (options Essai/affichage)	Quels types de jetons d'essai et d'affichage il convient que les compteurs à paiement ou le système de vente prennent en charge. Options: Une liste de jetons obligatoires et facultatifs est donnée en 6.3.8	6.3.8

Élément	Contexte	Référence
Power limit (c'est-à-dire: limite de puissance)	S'il convient, oui ou non, que les compteurs à paiement donnent une limitation de puissance et s'il convient, oui ou non, que le système de vente fournisse les jetons pertinents. Options: • il convient de mettre en œuvre la limite de puissance ou non; • établissement de la limite de puissance; • comment il convient que le compteur à paiement réagisse lorsque la limite de puissance est atteinte	6.2.4 6.3.9 8.6
Tariff rate (c'est-à-dire: taux de tarif)	Quelles sont les valeurs de taux de tarif pour les compteurs à paiement enregistrés dans la base de données du système de vente et si, oui ou non, il convient que le système de vente prenne en charge les jetons pertinents. Options: • préréglé par le constructeur; • variable et établi avec le jeton provenant du système de vente; • taux de tarif par compteur à paiement	6.2.6 6.3.11
Tamper detection (c'est-à-dire: Détection de falsification)	S'il convient, oui ou non, que les compteurs à paiement fournissent une détection de falsification et s'il convient, oui ou non, que le système de vente prenne en charge les jetons pertinents. Options: • il convient de mettre en œuvre la détection de falsification; • il convient de ne pas mettre en œuvre la détection de falsification; • il convient que le compteur à paiement prenne en charge le jeton de statut de falsification d'affichage • il convient que le système de vente prenne en charge le jeton de statut de falsification d'affichage. NOTE 3 Une prise en charge claire du jeton de falsification est obligatoire avec l'option 1	6.2.9
Phase power unbalance (Déséquilibre de puissance de phases)	S'il convient, oui ou non, que les compteurs à paiement donnent une limitation de déséquilibre de puissance de phases et s'il convient, oui ou non, que le système de vente fournisse les jetons pertinents. Options: • il convient de mettre en œuvre la limitation de déséquilibre de puissance de phases; • il convient de ne pas mettre en œuvre la limitation de déséquilibre de puissance de phases; • préréglé par le constructeur; • variable et établi avec le jeton provenant du système de vente; • la valeur de la limite de déséquilibre de puissance de phases; • comment il convient que le compteur à paiement réagisse lorsque la limite de déséquilibre de puissance de phases est atteinte	6.2.10 6.3.10 8.12
Initial credit (c'est-à-dire: Crédit initial)	Quelle valeur initiale il convient que le registre de crédit des compteurs à paiement ait lorsqu'il quitte les locaux du constructeur. Options: • vidé à zéro; • préréglé à une valeur initiale; • la valeur initiale	x
Special reserved TID (c'est-à-dire: TID réservé spécial)	S'il convient, oui ou non, que le système de vente mette en œuvre des identificateurs de jetons réservés spéciaux. Options: • il convient de ne pas mettre en œuvre des identificateurs de jetons réservés spéciaux; • il convient de mettre en œuvre des identificateurs de jetons réservés spéciaux; • détails spécifiés des identificateurs de jetons réservés spéciaux	6.3.5.2

Élément	Contexte	Référence
STS Certificate of Compliance (c'est-à-dire: Certificat de conformité STS)	Le fournisseur de produit conforme à la STS doit fournir une copie du certificat de conformité STS du produit particulier telle qu'elle a été émise par la CertificationAuthority compétente	C.10

C.12 Gestion du passage par zéro des TID

C.12.1 Introduction

Le Token Identifier (identificateur de jeton) est un champ de 24 bits, contenu dans le jeton conforme à la STS, qui identifie la date et l'heure de la génération du jeton. Il est utilisé pour déterminer si un jeton a été déjà utilisé dans un compteur à paiement. Le TID représente les minutes écoulées depuis le 1er janvier 1993. L'incrémentation du champ de 24 bits signifie qu'à un certain instant, la valeur de TID repasse à une valeur zéro.

Tous les compteurs de prépaiement STS sont affectés par le passage par zéro du TID le 24/11/2024. Tout jeton généré après cette date et utilisant le TID de 24 bits sera rejeté par les compteurs comme étant de vieux jetons car la valeur de TID incorporée au jeton aura été réinitialisée à 0.

Afin de surmonter ce problème, tous les compteurs exigent des jetons de changement de clé, le bit de passage par zéro étant mis. En outre, la date de référence du 01/01/1993 nécessite d'être changée pour être mise à une date ultérieure. Ce processus force les compteurs à réinitialiser la pile de TID à 0. Afin d'éviter que des jetons précédemment lus ne soient acceptés par le compteur du fait de la réinitialisation de la pile de TID, le processus de changement de clé doit introduire dans le compteur une nouvelle clé de décodeur.

Un processus est donc requis pour permettre la prise en charge de la gestion de ce changement avec l'impact le moindre sur les entreprises de distribution et les fournisseurs d'équipement.

Pour permettre la prise en charge d'une gestion plus facile des bases installées de grandes dimensions, il est proposé que la solution suivante gère le changement par compteur et pas par code de groupe d'alimentation (SGC), car certaines entreprises de distribution peuvent avoir une base installée de grandes dimensions sous un SGC unique.

C.12.2 Vue d'ensemble

C.12.2.1 Généralités

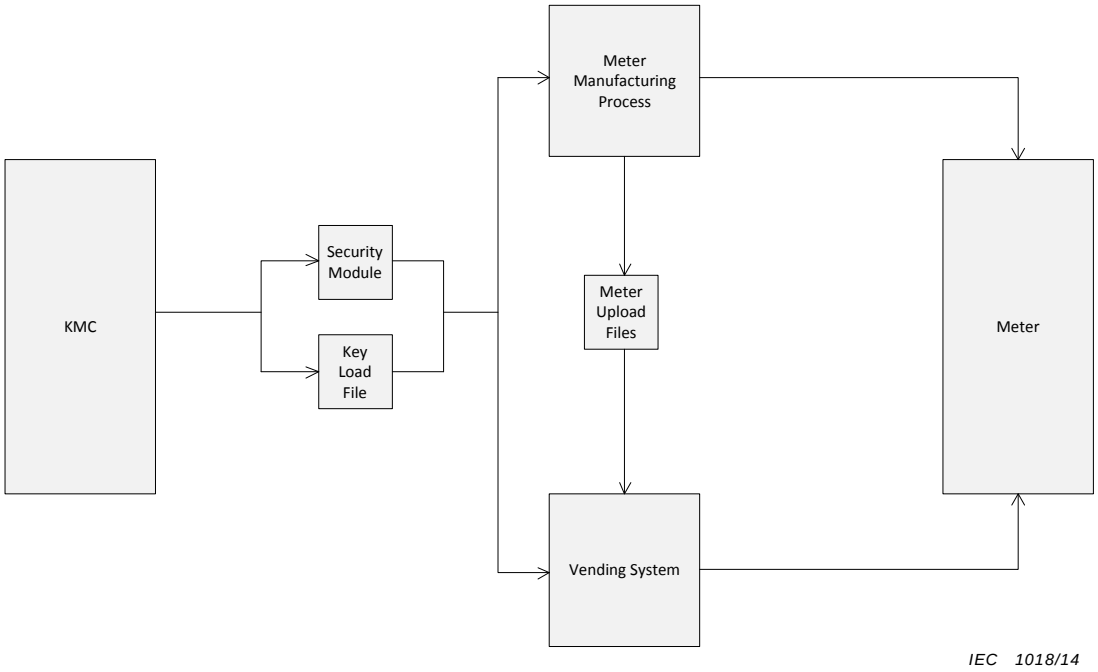
Les utilisateurs responsables de la gestion des compteurs à paiement doivent assurer l'adhésion à cette procédure par toutes les parties prenantes concernées.

Le problème actuel est que le TID est généré en utilisant une date de référence de 01/01/1993. La valeur de 24 bits atteint un point de passage par zéro le 24/11/2024. Afin de gérer ce phénomène, il est nécessaire de créer une nouvelle date de référence pour laquelle tous les jetons générés redémarreront avec une valeur de TID de 0. Bien que cela déplace en fait le problème, plus d'une date de référence à des intervalles décalés peut être utilisée.

Le présent Code de bonnes pratiques définit un processus pour gérer les clés de vente et les clés de décodeur sur la base de dates de référence différentes. Les éléments suivants, représentés à la Figure C.1, ont été inclus:

- Key management centre (Centre de gestion de clé);

- Security modules (Modules de sécurité);
- Vending systems (Systèmes de vente);
- Meter upload files (Fichiers de téléversement de compteur);
- Meter manufacturing equipment (Matériel de fabrication de compteur);
- Meters (Compteurs).



Légende

Anglais	Français
KMC	KMC (Centre de gestion de clé)
Security Module	Module de sécurité
Key Load File	Fichier de chargement de clé
Meter Manufacturing Process	Processus de fabrication de compteur
Meter Upload File	Fichier de téléversement de compteur
Vending System	Système de vente
Meter	Compteur

Figure C.1 – Vue d'ensemble du système

C.12.2.2 Centre de gestion de clé (KMC)

Le KMC est utilisé pour générer et charger des clés de vente (Vk) dans un module de sécurité. Le KMC génère également un fichier de chargement de clé (KLF) qui contient les données de chargement de clé pour un module de sécurité spécifique pour permettre à un système de vente de charger la Vk dans le module de sécurité attaché au système. Actuellement, le numéro de révision de clé (KRN) pour toute Vk est 1.

Afin de gérer la génération de jetons pour une date de référence spécifique, le système de vente exige que le KMC produise une nouvelle Vk pour chaque intervalle de date de référence. La nouvelle Vk est créée en incrémentant le KRN. Associée à chaque Vk dans le KLF, il y a la date de référence sélectionnée. Deux dates de référence sont prises en charge; à savoir le 01/01/2014 et le 01/01/2035. Seules deux sont requises, car il n'est pas envisagé

que les compteurs STS de la technologie actuelle soient encore en fonctionnement au moment où le TID de la Vk de 2035 repasse par zéro en 2066.

C.12.2.3 Module de sécurité

Le module de sécurité est requis pour générer des jetons de changement de clé, d'une Vk d'une date de référence donnée vers une Vk d'une nouvelle date de référence. Le firmware du SM est modifié pour permettre la mise en œuvre de la fonctionnalité du passage par zéro.

C.12.2.4 Système de vente

Le système de vente est requis pour gérer, avec chaque Vk chargée dans un module de sécurité, une date de référence associée. Cette date de référence est récupérée du fichier de chargement de clé généré au niveau du KMC. En outre, le système de vente doit associer chaque compteur enregistré à la date de référence de Vk à partir de laquelle il a été codé.

Une fois qu'une nouvelle Vk est rendue disponible, le système de vente doit permettre la gestion du processus de changement par lequel un compteur ou un groupe de compteurs peut être programmé pour un passage par zéro de la clé. De cette manière, les compteurs affectés subissent un changement de clé avec passage par zéro, ce qui réinitialise la pile de TID du compteur et génère une nouvelle clé de décodeur basée sur la nouvelle Vk. À partir de cet instant, tous les jetons générés pour le(s) compteur(s) sont chiffrés en utilisant la nouvelle Vk avec une valeur de TID calculée à partir de la date de référence correspondante.

Avec ce processus, les compteurs peuvent être programmés pour le changement de clé basé sur les exigences de l'entreprise de distribution. À tout instant, il y a deux clés de vente actives pour chaque SGC, car tous les compteurs n'ont pas leur clé changée à la nouvelle Vk en même temps.

C.12.2.5 Fichiers de chargement de compteur

Des compteurs neufs reçus en provenance des constructeurs peuvent être chargés dans le système de vente en utilisant un processus d'importation de fichier de téléversement de compteur. Ces compteurs sont codés par les constructeurs en utilisant une Vk avec la date de référence la plus récente et, donc, chaque enregistrement de compteur dans le fichier de téléversement de compteur est tenu d'inclure la date de référence pour laquelle il a été codé.

C.12.2.6 Matériel de constructeurs

Tous les compteurs partant de l'usine doivent être codés en utilisant une Vk avec l'actuelle (plus récente) date de référence, sauf accord contraire entre l'entreprise de distribution et le constructeur. Avec les deux dates de référence choisies, à savoir 2014 et 2035, tous les compteurs codés avant 2014 sont codés en utilisant la date de référence de 1993. Tous les compteurs codés entre 2014 et 2035 doivent être codés avec la date de référence de 2014 et tous les compteurs codés après 2035 utilisent une date de référence de 2035, sauf accord contraire entre l'entreprise de distribution et le constructeur.

C.12.2.7 Compteurs

Tous les compteurs conformes à la STS doivent prendre en charge le passage par zéro des clés.

C.12.3 Analyse d'impact

C.12.3.1 Généralités

Les domaines qui sont affectés par le processus ci-dessus sont énumérés ci-dessous:

C.12.3.2 Key management centre (Centre de gestion de clé)

- a besoin d'inclure une date de référence dans le fichier de chargement de clé pour chaque Vk;
- prend en charge la sélection des dates de référence prédéfinies pour générer la Vk;
- les modules de sécurité doivent prendre en charge le fanion de passage par zéro des clés.

C.12.3.3 Vending systems (Systèmes de vente)

- associent chaque Vk pour un SGC à une date de référence dérivée à partir du fichier de chargement de clé généré par le KMC;
- associent chaque compteur à une date de référence pour laquelle le compteur est codé. Cela doit inclure l'extraction de la date de référence dans l'importation de fichier de téléversement de compteur;
- autorisent les compteurs associés à une date de référence antérieure à programmer individuellement, dans des groupes ou par le SGC pour un changement de clé en Vk à une nouvelle date de référence et incluent le fanion de passage par zéro des clés.

C.12.3.4 Fichiers de chargement de compteur

- Les constructeurs ont à inclure une date de référence avec chaque enregistrement de compteur dans le fichier;
- Les spécifications du fichier de téléversement de compteur ont besoin d'être mises à jour pour refléter l'ajout de la date de référence.

C.12.3.5 Matériel de fabrication

- Doit coder automatiquement tous les compteurs utilisant la Vk avec la date de référence active la plus récente selon l'accord conclu avec l'entreprise de distribution;
- Meters (Compteurs);
- Doivent prendre en charge le passage par zéro des clés.

C.12.4 Dates de référence

Voir 6.3.5 ci-dessus.

C.12.5 Mise en œuvre

C.12.5.1 Généralités

Les détails de mise en œuvre pour les constructeurs de compteurs et les systèmes de vente ont été présentés dans les grandes lignes dans le corps de document. Les sections qui suivent donnent les lignes directrices de base que les entreprises de distribution suivent dans la mise en œuvre réussie du programme de changement de clé TID. Noter que des entreprises de distribution peuvent choisir de suivre des méthodes alternatives de mise en œuvre.

C.12.5.2 Hypothèses

Avant de démarrer la mise en œuvre des changements de clé sur le terrain, il est présupposé que les éléments suivants ont été achevés par les constructeurs de compteurs, les systèmes de vente, et les modules de sécurité:

a) Le firmware de module sécurisé a été changé pour prendre en charge la fonctionnalité de passage par zéro.

b) Les fournisseurs de logiciel de vente ont modifié le logiciel de vente pour reconnaître les dates de référence décrites dans la présente norme. Une fois qu'un compteur a eu sa clé changée avec passage par zéro, ce fait doit être enregistré dans la base de données de vente.

c) Tous les compteurs manufacturés prennent en charge la fonctionnalité de passage par zéro telle que spécifiée dans l'IEC 62055-41. Lorsque tel n'est pas le cas, il faut remplacer les compteurs par des compteurs qui prennent en charge la fonctionnalité de passage par zéro. Il est envisagé que la base installée actuellement ne sera plus en service au moment où le passage par zéro des clés sera requis et que tous les compteurs manufacturés après le premier changement de date de référence de 2014 prendront en charge la fonctionnalité de passage par zéro.

C.12.5.3 Processus pour les entreprises de distribution

Un guide pour le processus à suivre est présenté ci-dessous:

a) Planifier le programme de passage par zéro de TID afin de parachever la base installée de compteurs au moins une année avant la date critique du 24/11/2024.

b) Communiquer le plan, et les raisons du programme, à toutes les régions au sein de l'entreprise de distribution.

c) Mettre à niveau toutes les installations de vente au logiciel qui prend en charge la fonctionnalité de passage par zéro et les changements appropriés de base de données.

d) Mettre à niveau le logiciel d'entreprise de distribution pour assurer qu'il prend en charge les nouveaux formats neufs fichier de téléversement de compteur, lorsqu'ils sont utilisés comme outil d'importation.

e) Mettre à niveau/acheter des modules sécurisés avec la fonctionnalité de passage par zéro par l'intermédiaire du fournisseur de module sécurisé.

f) Mettre à niveau le logiciel du KMC, lorsqu'il est la propriété d'une entreprise de distribution, pour permettre les dates de référence multiples.

g) Entrer en contact avec le constructeur de vos compteurs pour confirmer si, oui ou non, ses compteurs prennent en charge le changement de clé avec passage par zéro. Si la réponse est non, ces compteurs sont à remplacer sur le terrain par des compteurs qui le font.

h) Démarrer le processus de changement de clé.

C.12.5.4 Processus de changement de clé

Il existe diverses options pour le processus de changement de clé physique:

a) Générer des jetons de changement de clé (deux jetons) pour une région et envoyer des techniciens sur le terrain pour insérer systématiquement ces jetons dans chaque compteur inspecté.

b) Générer (automatiquement) des jetons de changement de clé lorsqu'un achat de crédit est effectué par l'utilisateur. Expliquer à l'utilisateur que le jeton de crédit ne fonctionnera pas si les jetons de changement de clé n'ont pas tout d'abord été introduits dans le compteur. Typiquement, cela constitue déjà la pratique standard pour les changements de clé.

c) Communiquer le programme aux utilisateurs finaux et leur demander de venir chercher leurs jetons de changement de clé au plus tard à certaines dates limites.

Toutes les options ci-dessus ont des avantages et des inconvénients.

L'option a) s'assure que les changements de clé sont effectués de façon systématique par zone, qui peut alors être "cochée" comme étant achevée. Elle est maîtrisable mais coûteuse en main-d'œuvre.

L'option b) est de loin moins coûteuse, mais elle ne permet pas que les régions ou les zones soient réalisées d'une manière maîtrisée car on ne peut pas être sûr que les jetons ont été introduits tant qu'un nouvel achat n'a pas été réalisé. Cette option ouvre également la possibilité que de nombreuses plaintes soient reçues pour des jetons de crédit non fonctionnels si ces jetons sont introduits sans l'introduction préalable de jetons de changement de clé.

L'option c) est la moins souhaitable, car la communication du problème va directement à l'utilisateur final et peut causer des soucis inutiles.

C.12.5.5 Communication du programme

Ci-dessous est présenté un guide montrant la forme possible que peut prendre la communication aux bureaux régionaux des entreprises de distribution. Noter qu'il s'agit d'un guide seulement, susceptible d'être changé pour être adapté aux préférences d'entreprises individuelles de distribution selon les besoins.

Adresses et en-têtes appropriés

Objet: Programme de changement de clé de compteur sur le terrain

Comme on peut le savoir, tous les compteurs de prépaiement stockent des jetons introduits comme un moyen d'amener un compteur à cesser d'accepter un jeton qui a déjà été utilisé. En plus de ce stockage, chaque jeton a également, intégré en 20 chiffres, la date et l'heure auxquelles le jeton a été généré. Le compteur compare ensuite cette date et cette heure au plus ancien jeton dans sa mémoire et rejette le jeton s'il est plus ancien que le jeton le plus ancien dans cette mémoire.

Le champ date et heure du jeton a une plage maximale de 31 ans. Cela signifie qu'après 31 ans d'incrémentation de ce champ date et heure, la valeur stockée "passera" par zéro – à la manière d'un compteur kilométrique de voiture "faisant un tour complet".

Le jeton courant "passera" en novembre 2024 à la date de départ actuelle de 1993. À ce moment, la date et l'heure sur les jetons passeront à sa date zéro (1993), point auquel les compteurs n'accepteront plus les jetons générés avec cette date de référence.

Alors que la date de 2024 peut sembler être un long terme dans le futur, on a besoin de planifier le changement de cette date de référence de 1993 pour la faire passer à une date de référence postérieure. À cet effet, des constructeurs ont été sensibilisés au fait que des changements sont à apporter aux compteurs, aux Modules de sécurité, aux systèmes de vente et aux Centres de gestion de clé pour prendre en charge ce changement.

Le changement consiste à changer la clé dans chaque compteur sur le terrain, ce qui peut être réalisé en remettant un ensemble de jetons de changement de clé à l'utilisateur ou en mettant en œuvre un programme par lequel chaque compteur est inspecté par une équipe technique pour introduire ces jetons.

Afin de réduire le nombre de compteurs qui sont à inspecter ou voir leur clé changée, sur le terrain, les constructeurs sont instruits du fait que tous les compteurs fabriqués à partir de 2014 doivent être codés de la nouvelle date de référence de 2014. Cela signifie qu'il convient que le nombre réel de compteurs avec une date de référence de 1993 soit réduit de façon drastique avant 2024 et que peu de compteurs nécessitent des changements de clé.

Avec les systèmes actuellement envisagés par la STS Association, il convient que ce processus n'ait jamais à être répété, car la date de référence des compteurs changera tous les 21 ans.

Bibliographie

ISO 8731-1, *Banque – Algorithmes approuvés pour l'authentification des messages – Partie 1: DEA*

ISO 4909, *Cartes d'identification – Cartes de transactions financières – Contenu des données de plage magnétique pour la piste 3*

ISO/IEC 7498-1, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base: Le modèle de base*

ISO/IEC 9545, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Structure de la couche application*

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 201-15.1.0, *Standard transfer specification (STS) – Companion specification – Meter function object: RegisterTable for electricity payment meters*

FIPS PUB 198, *The Keyed-Hash Message Authentication Code (HMAC)*

FIPS PUB 197, *Advanced Encryption Standard*

FIPS PUB 186-2, *Digital Signature Sta*ndard

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*

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)*

NOTE Les documents STS sont disponibles sur le site web de la STS Association sous www.sts.org.za

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch