IEC 62055-42

Edition 1.0 2022-07

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Electricity metering – Payment systems – Part 42: Transaction Reference Numbers (TRN)

Comptage de l'électricité – Systèmes de paiement – Partie 42: Numéros de référence des transactions (TRN)

IEC 62055-42:2022-07(en-fr)

THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2022 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

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

Tel.: +41 22 919 02 11
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 corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
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 It also gives information on projects, replaced and withdrawn publications.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 once a month by email.

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: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch
Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

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.

Recherche de publications IEC - webstore.iec.ch/advsearchform
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 une fois par mois par email.

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

IEC Products & Services Portal - products.iec.ch
Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org
Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 300 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 19 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

IEC 62055-42

Edition 1.0 2022-07

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Electricity metering – Payment systems – Part 42: Transaction Reference Numbers (TRN)

Comptage de l'électricité – Systèmes de paiement – Partie 42: Numéros de référence des transactions (TRN)

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

ICS 17.220.20, ICS 35.100.70, ICS 91.140.50

ISBN 978-2-8322-3951-3

Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.

® Registered trademark of the International Electrotechnical Commission
Marque déposée de la Commission Electrotechnique Internationale

CONTENTS

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

ELECTRICITY METERING – PAYMENT SYSTEMS –

Part 42: Transaction Reference Numbers (TRN)

FOREWORD

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

IEC 62055-42 has been prepared by IEC technical committee 13: Electrical energy measurement and control. It is an International Standard.

The text of this International Standard is based on the following documents:

Draft	Report on voting
13/1843/CDV	13/1860/RVC

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

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

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

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

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

IMPORTANT – The "colour inside" logo on the cover page of this document indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

INTRODUCTION

The IEC 62055 series recognizes and takes into account the concept of layered interoperability for use within the smart metering and smart grid domains.

It also ensures system element interoperability above the semantic layer to include business function and business process interoperability layers within an electricity metering system, thus ensuring overall compatibility at all these levels.

This document is based on the principles the IEC 62055 standards are built on and sets the rules for future extensions to guarantee consistency, thus providing a common vocabulary for use by utilities to express requirements in tenders and also by vendors to have a unified understanding for interpretation of the tender requirements.

This document forms part of the IEC 62055 series and shares some references with IEC 62055-41, in that both standards represent TransferCredit tokens utilising 20-digit token carriers. However, IEC 62055-41 and IEC 62055-42 differ greatly in their encoding, security mechanism and intended use cases. Whereas IEC 62055-41 is meant for predominantly offline systems, IEC 62055-42 is intended for mostly online systems where the decimal token carrier is used as a back-up mechanism for vending while meters are intermittently offline.

The IEC 62055 series has been developed by IEC TC13 specifically for electricity metering systems, but it is equally applicable in the domain of other utility services such as water and gas.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

ELECTRICITY METERING – PAYMENT SYSTEMS –

Part 42: Transaction Reference Numbers (TRN)

1 Scope

This document specifies a token generation mechanism and token structure for smart prepayment functionality in markets where IEC 62055-41 compliant systems are not used, and where a different security mechanism is required by project-specific or national requirements. This document specifies token structure, authentication and an anti-replay mechanism, token operating model, and protocol.

This document is informed by the STS Association key management services, and by the key management mechanisms used within the DLMS/COSEM security model within IEC 62056-6-2. Reference is made to the international STS token standards (IEC 62055-41, IEC 62055-51 and IEC 62055-52) for payment metering systems, and interworking has been considered where appropriate in terms of token carrier ranges in the decimal domain. IEC 62055-41 tokens and those described in this document are not interoperable, however their domains are designed to be mutually exclusive to ensure the two kinds of tokens do not interfere with each other.

Metering application processing and functionality, HAN interface commands and attributes, WAN interface commands and attributes are outside the scope of this document; however, reference is made to other standards in this regard.

The mechanism for auditing and retrieving data from the meter relating to tariffication, meter readings, profile data and other legal metrology information is outside the scope of this document; however, this is defined as part of any overall metering solution. Such interfaces for retrieving data from a meter may be defined using suitable protocols such as DLMS/COSEM as defined in the IEC 62056 series.

2 Normative references

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

IEC 60050-300:2001, *International Electrotechnical Vocabulary (IEV) – Part 300: Electrical and electronic measurements and measuring instruments – Part 311: General terms relating to measurements – Part 312: General terms relating to electrical measurements – Part 313: Types of electrical measuring instruments – Part 314: Specific terms according to the type of instrument*
IEC 60050-300:2001/AMD1:2015
IEC 60050-300:2001/AMD2:2016
IEC 60050-300:2001/AMD3:2017
IEC 60050-300:2001/AMD4:2020

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

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

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

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

IEC 62055-41:2018, *Electricity metering – Payment systems – Part 41: Standard transfer specification – Application layer protocol for one-way token carrier systems*

IEC 62056-5-3:2017, *Electricity metering data exchange – The DLMS/COSEM suite – Part 5-3: DLMS/COSEM application layer*

IEEE EUI 64, https://standards.ieee.org/develop/regauth/tut/eui64.pdf

Verhoeff, J.,1975, *Error Detecting Decimal Codes,* (Tract 29)

NIST SP 800-38D: 2007, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*

3 Terms, definitions, abbreviated terms and notation

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 60050-300:2001, IEC TR 62051:1999, IEC 62055-31:2005, IEC 62055-41:2018, IEC 62055-21:2005 and the following apply.

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

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

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

3.1.1
companion specification
regional, consortia, or manufacturer-specific set of requirements and choices taken from a base standard, in order to facilitate a particular set of use cases

3.1.2
meter
measuring instrument; synonymous with “payment meter” and “decoder” where the decoder is a sub-part of of a multi-device installation

3.1.3
POS
point of sale; entity that may create and transfer tokens, synonymous with “CIS”, “MIS” and “HHU”

3.1.4
super token
group of digits derived from more than one data block of the APDU that shall be entered together in sequence to form a larger message

3.1.5
utility
retailer or supplier of an energy or water commodity service

Note 1 to entry: 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.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

3.2 Abbreviated terms

AES	Advanced Encryption Standard
AAD	AdditionalAuthenticatedData
AMT	Amount
APDU	ApplicationProtocolDataUnit
CIS	Customer Information System
COSEM	Companion Specification for Electricity Metering
DLMS	Device Language Message Specification
EUI	Extended Unique Identifier
GMAC	Galois Message Authentication Code
HAN	Home Area Network
HES	Head End System
HHU	Hand Held Unit
HMI	Human to Machine Interface
ID	Identifier
IHD	In-Home Display
IEEE	Institute of Electrical and Electronics Engineers
KMS	Key Management System
LCS	Load Control Switch
LSB	Least Significant Byte
MAC	MessageAuthenticationCode
MIS	Management Information System
MOD	Modulo operation
MS	Most Significant
MSB	Most Significan Byte
NIST	National Institute of Standards and Technology
NSP	National Service Provider
PIN	Personal Identification Number
POS	Point Of Sale
RANDz	Reasonable And Non-Discriminatory with Zero royalty
RES	Reserved
SE	Smart Energy
SMS	Short Message Service
STN	SequentialTokenNumber
STS	Standard Transfer Specification
TC	Technical Committee
TCDU	TokenCarierDataUnit
TMAC	Truncated Message Authentication Code
TOU	Time Of Use
TSTN	TruncatedSequentialTokenNumber
WAN	Wide Area Network
WG	Working Group

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

3.3 Notation

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.

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.

The symbol "||" inserted between two expressions means that the resultant values of the two expressions are concatenated into a single string without separators.

4 Numbering conventions in this document

Numbers are in decimal format, unless otherwise indicated. Decimal digit values range from 0 to 9.

Binary digit values range from 0 to 1. Binary numbers are indicated by suffix "b". The representation of binary strings is such that the least significant bit is to the right of the string representation and the most significant bit is to the left of the string. Numbering of bit positions starts with bit position 0 which corresponds to the least significant bit of the binary number.

Hexadecimal digit values range from 0 to 9 and A to F and are indicated by prefix "0x" or suffix "hex".

5 Reference smart meter model

5.1 Generic functional reference diagram

In the general case any meter designed to be compliant with this document may contain some or all of the functions specified in Figure 1 and a minimum of 1 credit and 1 charge type (more are permitted). The diagram in Figure 1 is provided here for reference and context, but is outside the normative scope of this document.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

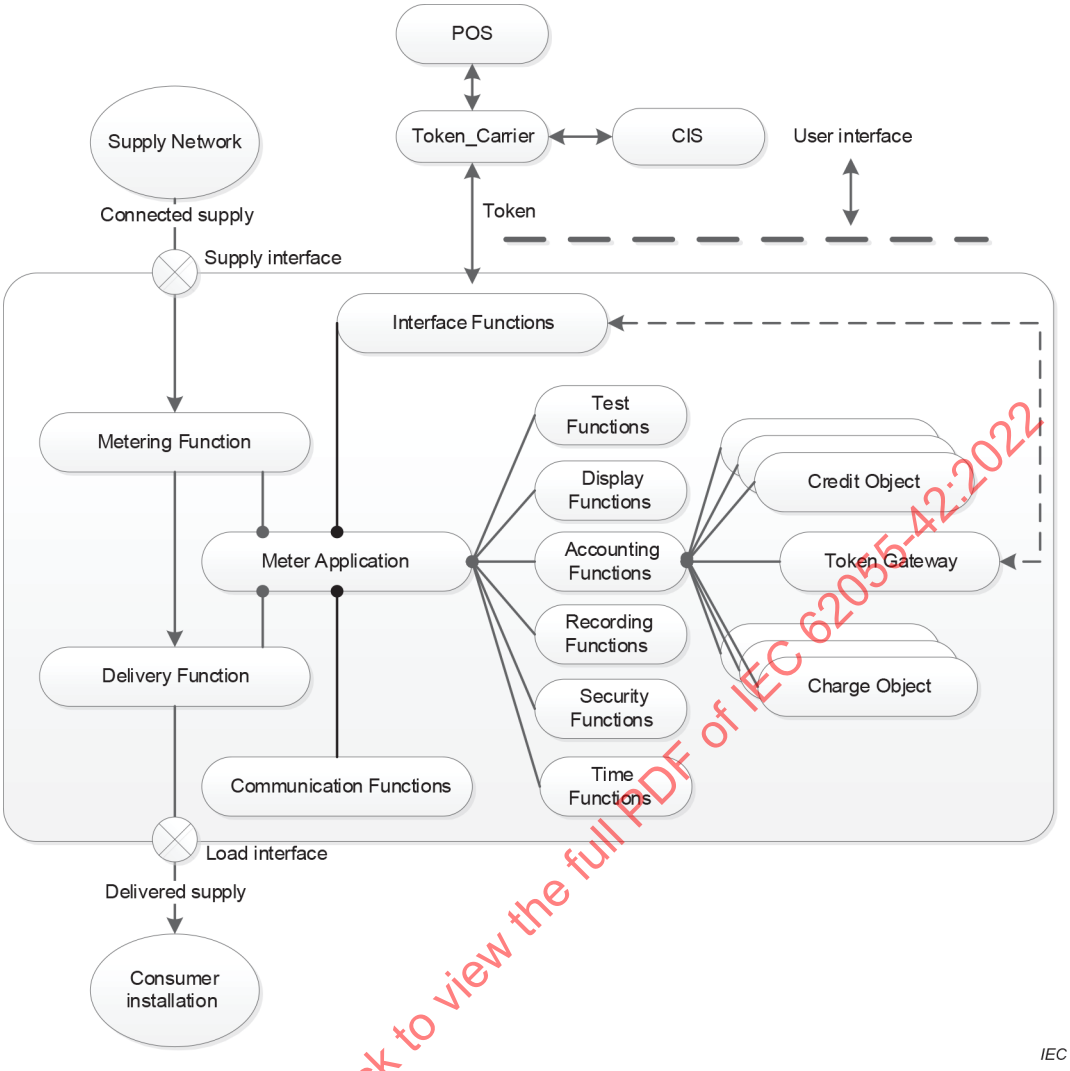


Figure 1 – Functional block diagram of a generic payment meter

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

This document primarily deals with the structure, generation, transport and security of the token itself as shown in Figure 1.The generation of the tokens, in the case of meter-terminating tokens, happens within some form of software system outside the meter domain by using information that is specific to a meter and the generating party. In the case of meter-originated tokens the generation of tokens happens within the meter using information relating to the specific meter and the intended recipient of such tokens. This information contains, but is not limited to, meter identification information and security material such as encryption keys and authentication keys.

The model represents a typical payment meter, housing all the above functions; however, it is possible that other multi-device architectures may comply with the generic model. For example, delivery of the token could take place through a proxy device.

It is also possible for a token in this architecture to be delivered through the HAN or WAN interface or separate Human Machine Interface (HMI).

Because of the variety of use cases required for smart metering it may be necessary for a token to be transported / tunnelled through multiple protocols. This means that a common transport protocol is not practical and is not necessary for this document.

All that is required for it to be transported is to define the data format and payload of the token and ensure that the format of the token is simple and flexible enough to be accommodated as a tunnelled object within any transport protocol (from the view of the token).

5.2 Token transfer protocol reference model

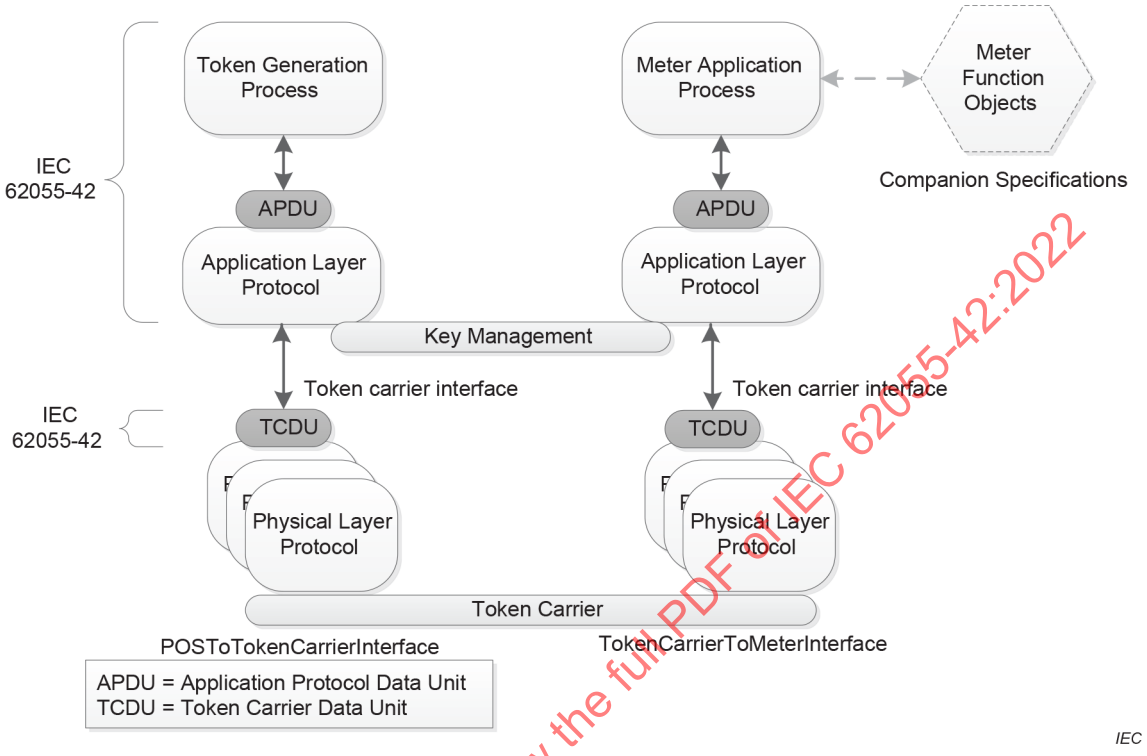


Figure 2 – Reference model as a 2-layer collapsed OSI protocol stack

Figure 2 shows the token generation system and meter interface using a protocol representation describing a system that generates a token using some specific concatenated data items to make a binary payload, which is optionally encrypted (depending on SubClass), creating an ApplicationProtocolDataUnit (APDU). The binary payload is then converted into a decimal number that is known as the TokenCarrierDataUnit (TCDU) and the TCDU can be transported to a meter in some way via a physical protocol layer (this could be via a paper printed token carrier, over the air or any other mechanism). Where human interaction is required (e.g. when printed, entered via a keypad, or spoken) the TCDU shall be in decimal form but may be represented in a format other than decimal, e.g. over the air transport. After transporting the TCDU to the meter's interface, it is the responsibility of the meter's application protocol layer to convert and decrypt the TCDU and deliver the payload to the appropriate function in the payment meter application process.

The TCDU may be delivered to the meter in a variety of ways, including but not limited to:

- remotely by some communications interface ("over the air");
- locally by a physical token carrier;
- manual entry process initiated by the user.

In addition to a TCDU created for delivery to a meter it is also possible within this document for the meter to create a TCDU for delivery to and processing by some other external software system (see 6.2.5.4).

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

5.3 Dataflow from the POSApplicationProcess to the TokenCarrier

Figure 3 shows a generic process for creating a TCDU from an APDU, whereby binary data items are concatenated to construct an APDU and then undergo a number of mandatory and optional operations to convert the APDU into a decimal TDCU. The TCDU is then used as the encoded payload to send to a payment meter over a variety of physical and transport media.

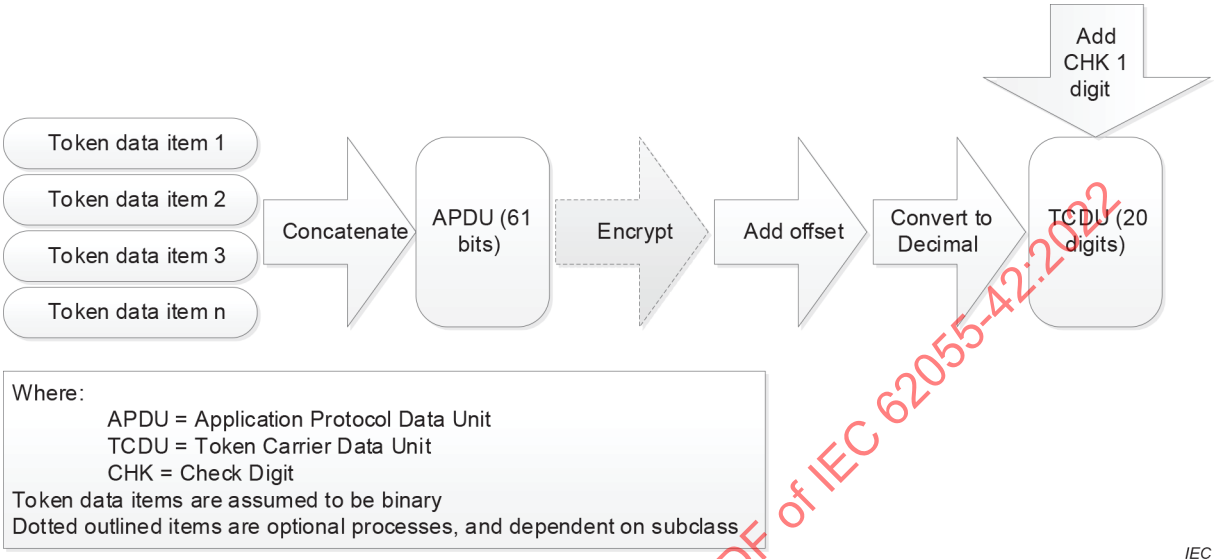


Figure 3 – Generic model of POSApplicationProcess to TokenCarrier

5.4 Dataflow from the TokenCarrier to the MeterApplicationProcess

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

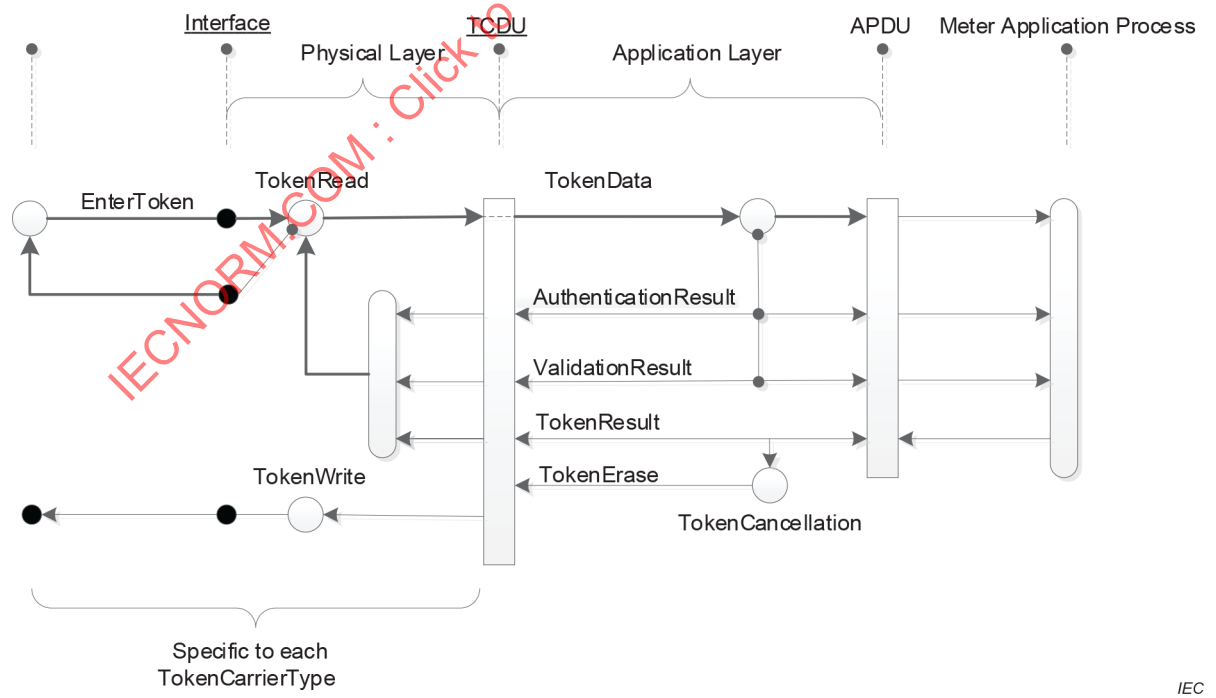


Figure 4 – Dataflow from the TokenCarrier to the MeterApplicationProcess

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

5.5 MeterFunctionObjects / companion specifications

As shown in Figure 2, the TokenCarrierToMeterInterface, which also includes the TokenCarrier, is dealt with in this document. The remaining MeterFunctionObjects shown in the diagram are defined in companion specifications and are not normative to this document. (See 3.1.1).

6 POSToTokenCarrierInterface application layer protocol

6.1 APDU: ApplicationProtocolDataUnit

6.1.1 Data elements in the APDU

The Token payload in the APDU is a (61-bit for SingleTokenPayload) binary payload comprising a number of smaller data elements as given in 6.3.

To construct a full APDU, additional data elements are required such as SubClass, Truncated SequentialTokenNumber (TSTN), Data and TruncatedMessageAuthenticationCode (TMAC).

Table 1 gives details of Basic and Derived data for elements of the APDU construction.

Table 1 – Basic and derived elements of APDU and TCDU construction

Data element name	Context	Data element format	Size	Range	Basic/ Derived	Reference
AdditionalAuthentication Data (AAD)	Additional octets of data that can be placed in the MAC calculation string for calculation of a MAC.	Binary	Variable	Variable	Derived	6.1.14
Data	Data in APDU depends on SubClass type	Binary	Variable	All possible values based on data size	Basic or Derived data depending upon SubClass type	
FunctionIndex	A parameter which is used to enforce token acceptance at receiving ed in a particular order.	Binary	32 bits	All possible	Basic	6.1.9
InitializationVector (IV)	Part of the AAD required for the GMAC algorithm.	Binary	96 bits	Fixed value of 0x00000000 concatenated with SupplierID of 64 bits	Derived	Figure 7
MessageAuthenticationCode (MAC)	Message authentication code generated using the TokenPayload and other data elements. This MAC is then truncated to form TruncatedMAC and the TMAC added to the TokenPayload as the LSB to form the APDU	Binary	128 bits	All possible values	Derived	6.1.13
MessageIdentifier	It is an octet data that is used to prepare AAD and is made up of multiple data elements as per SubClass type.	Binary	Variable	Variable	Derived	6.1.5
MeterID	Individual unique identification for a specific meter.	Binary	64 bits		Basic	6.1.3

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Data element name	Context	Data element format	Size	Range	Basic/ Derived	Reference
Offset	A constant numerical value used in production of a TCDU to offset the numerical token to a value that does not clash with Class 0, 1, 2, 3, 4 or 6 tokens.	Binary or decimal according to context of use	63 bits or 19 digits	Specific value detailed in Table 9		6.1.17
SequentialTokenNumber (STN)	A per meter unique token number that is used in full to create a MAC for an APDU, of which only the 10 LS bits are included in the APDU and referred to as the TruncatedSequentialTokenNumber (TSTN)	Binary	32 bits	All possible values	Basic	6.1.6
SingleTokenPayload	The actual token data that is to be transferred to the meter prior to processing, including truncated MAC and/or encryption, with 3 additional MS bits 000b to pad out to 64 bits.	Binary	61 bits	All possible values	Derived	6.1.11
SuperTokenPayload	The payload considered from multiple tokens when a series of tokens is presented to the meter in order. This payload is larger than the 61 bits as it shall incorporate the initial 61-bit payload of the first data block in the series (with 3 additional MS bits with 000b to pad out to 64 bits) and then all of the binary payloads of the subsequent data blocks in the set. Also for each data block 2 onwards in the set there shall be in the inclusion of 3 additional bits of 000b value to pad to 64-bit equivalent payloads.	Binary	122 bits Or 183 bits Or 244 bits	3 discrete values depending on whether the SuperToken is 2 or 3 or 4 blocks long.	Derived	6.1.12
SubClass	Defines the type of token and whether token is encrypted or not. This information is never encrypted in token	Binary	4 bits	All possible values	Basic	6.2.3 and 6.2.4
SuperTokenBlockToFollow	Identification of the number of the block in the chain of data blocks in the case of SubClass 10 token. This value represents the number of blocks to follow.	Binary	2 bits	All possible values (00b, 01b, 10b) except 11b		See B.1.2
SupplierID	An identification number specific to the party creating the token.	Binary	64 bits	All possible values	Basic	6.1.2
TruncatedSequnetialTokenNumber (TSTN)	As STN data size is 32 bits, its 10 LS bits, referred to as TruncatedSequentialTokenNumber (TSTN), are used in the APDU payload data.	Binary	10 bits	All possible values	Derived	6.1.7

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Data element name	Context	Data element format	Size	Range	Basic/ Derived	Reference
TruncatedMAC (TMAC)	32 LS bits of MAC	Binary	32 bits	All possible values	Derived	6.1.13
TokenOriginationID	An indication of the direction of the token – meter originating or meter terminating.	Binary	8 bits	0x01, 0x02	Basic	6.1.4

6.1.2 SupplierID

The SupplierID uniquely identifies the creater of the token that may be a utility, vending network, or some other third party service provider. The SupplierID may be a project-specific value assigned to an individual meter and shall be globally unique. Typically, this could be a client System Title as defined in DLMS/COSEM (See IEC 62056-5-3:2017, 4.1.3.4) or an IEEE EUI 64 or some other 64-bit identifier chosen for a project and managed by some official Registration Authority or other party to guarantee its uniqueness.

6.1.3 MeterID

The MeterID may be a project-specific value assigned to an individual meter and shall be globally unique. Typically this could be a meter (server) System Title as defined in DLMS/COSEM (See IEC 62056-5-3:2017, 4.1.3.4) or an IEEE EUI 64 or some other 64-bit identifier chosen for a project and managed by some official Registration Authority or other party to guarantee its uniqueness.

6.1.4 TokenOriginationID

This field indicates whether the token was generated from the meter or from the HES and is included in the MAC calculation. The assigned values are:

- 0x01 for Commands (to meter);
- 0x02 for Responses (from meter);
- 0x03….0xFF reserved for future assignment.

6.1.5 MessageIdentifier

It is an octet data array which is used in the derivation of AAD. MessageIdentifier data derivation information for unencrypted and encrypted sub-classes is given in Table 2.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table 2 – SubClass-wise MessageIdentifier detail and SubClass Functional Class

SubClass	Description	Message-Identifier	SubClass Functional Class	Remarks
Unencrypted sub-classes				
0	Unencrypted TransferCredit	SupplierID ||MeterID ||TokenOriginationID (0x01) ||STN (Supplier generated non- SpecialFunction token counter) ||FunctionIndex (optional)	1	FunctionIndex will remain 0 as no "TransferCredit + Function" token SubClass is present
1	SpecialFunction token	SupplierID ||MeterID ||TokenOriginationID (0x01) ||STN (Supplier generated SpecialFunction token counter)	3	
2 – 7	RESERVED			
Encrypted sub – classes				
8	TransferCredit	SupplierID ||MeterID ||TokenOriginationID (0x01) ||STN (Supplier generated non-SpecialFunction token counter) ||FunctionIndex	1	FunctionIndex will be the value as used in previously issued "TransferCredit + Function" token.
9	SpecialFunction token	SupplierID ||MeterID ||TokenOriginationID (0x01) ||STN (Supplier generated SpecialFunction token counter)	3	Whenever a SpecialFunction token is accepted by a meter then any earlier SpecialFunction tokens will be rejected if entered subsequently.
10	TransferCredit + Function	SupplierID ||MeterID ||TokenOriginationID (0x01) ||SequentialTokenNumber (Supplier generated non-SpecialFunction token counter) ||FunctionIndex	1	FunctionIndex = (last FunctionIndex + 1)
11	MeterGenerated token	SupplierID ||MeterID ||TokenOriginationID (0x02) ||SequentialTokenNumber (MeterGenerated token counter)	No SubClass group rule required for token to be accepted at HeadEndSystem.	
12 – 15	RESERVED			

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.1.6 SequentialTokenNumber (STN)

This value is a 32-bit number that is generated and managed by the HES (or by the meter in the case of meter-generated tokens). It is a monotonically increasing counter that increments with each token generated for, or by, the meter in order to prevent replay of tokens. Each token generated for, or by, a particular meter shall have a different STN until the point at which the STN rolls over. However, it is assumed that with normal TransferCredit patterns, the periodicity of roll over will be very long (much greater than the lifetime of a meter). The STN shall be truncated and the 10 least significant bits shall be used in the token payload and shall be called the TSTN (see 6.1.7); however, the full 32-bit value of STN, which is reconstructed by the meter, shall be used to calculate the MAC for the token.

The generation of a token for or by any one meter shall not affect the sequence of STN instances generated for or by any other meter. The first token generated for any SubClass shall have an STN value of 0x00000001.

6.1.7 TruncatedSequentialTokenNumber (TSTN)

To save space within the token the full 32-bit STN is not transferred. Its MS 22 bits change very infrequently and can safely be deduced from the 10 LS bits of a token; these LS 10 bits are the TSTN.

There are various ways to deduce the full STN based on the TSTN, one of which is by the principle of a sliding window whereby the meter (or the HES in the case of a meter-generated token) shall never accept a token that has a TSTN that is more than X higher or more than Y lower than the last accepted token where the unsigned sum of X and Y is no more than 256. The reason for this is to prevent the unobserved rollover of the 10-bit field of STN (i.e. TSTN). Given that older tokens are more likely to be presented to the meter out of order than newer tokens, it is recommended that the sliding window of token acceptance is biased towards the past, meaning that Y will be larger than X.

6.1.8 Deducing the MS part of STN and validating TSTN

STN, N, is validated by relating it to previous STN values accepted by the meter (or HES in the case of a meter-generated token) for tokens of the same SubClass.

The following data need to be used by the meter (HES in case of a meter generated token) to deduce and validate STN for each SubClass:

a) M_N is the maximum value of N of the token already accepted (by the meter or HES as per SubClass) where M has an initial value of 0 (unless otherwise specified in a companion specification);

b) L_N is the lower acceptable value of N that is valid, where L_N has an initial value of 1 for all sub-classes;

c) U_N is the upper acceptable value of N that is valid (having initial value of 128 for SubClass 0, 128 common value for SubClass 8 and SubClass 10, whereas intial value is 512 for SubClass 9 as derived from Table 3;

d) $LS(N)$ (i.e. TSTN) is the least significant part of N, which is available in the token payload;

e) R is the number of possible values of TSTN. For example: TSTN has a 10-bit field and R has a value of 1 024 and TSTN has possible values from 0 to 1 023;

f) $MS(N)$ which is included as input information in MAC and can be deduced at the meter end;

g) M_N, limits L_N and U_N are stored having data size similar to the full STN value of 32 bits and are not transmitted in the token.

Table 3 gives one example of defining L_N and U_N for each SubClass of token considering the number of tokens to be issued.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table 3 – Example of defining L_N and U_N for each SubClass

Sub Class	L_N	U_N	Comment
0	M_N + 1 – 3R/8 or L_N if value of (M_N + 1 – 3R/8) is less than present value of L_N	M_N + R/8	The range between, and including the limits, is normally R/2. This defines a window of acceptance. The upper limit is a few more than the highest transaction number accepted so that some future transactions can be entered out of sequence. L_N shall be stored. U_N can be calculated from M_N.
1	M_N + 1	M_N + R/2	Only transactions later than the last one accepted will be accepted, but there is a range of tolerance for omitting transactions from the sequence. L_N and U_N can be calculated from M_N.
8 and 10	M_N + 1 – 3R/8 or L_N if value of (M_N + 1 – 3R/8) is less than present value of L_N	M_N + R/8	The range between, and including the limits, is normally R/2. This defines a window of acceptance. The upper limit is a few more than the highest transaction number accepted so that some future transactions can be entered out of sequence. L_N shall be stored. U_N can be calculated from M_N. SubClass 10 token acceptance in strict sequence of issuance by POS/HES is ensured by the parameter called "FunctionIndex". This is described in 6.1.9.
9	M_N + 1	M_N + R/2	Only transactions later than the last one accepted will be accepted, but there is a range of tolerance for omitting transactions from the sequence. L_N and U_N can be calculated from M_N.
11	M_N + 1 – 3R/8 or L_N if value of (MN + 1 – 3R/8) is less than present value of L_N	M_N + R/8	The range between, and including the limits, is normally R/2. This defines a window of acceptance. The upper limit is a few more than the highest transaction number accepted so that some future transactions can be entered out of sequence. L_N shall be stored. U_N can be calculated from M_N.
X …			X is future SubClass which may use additional methods to enforce different rule/rules. New sub-classes could be defined with the same behaviour as existing ones; this could allow distinct sub-classes to use identical acceptance rules without their numbers interacting as would be the case if they were assigned to the same sub-class.
NOTE For SubClasses 8 and 10 only 1 row is used in above column. This is to indicate that these two sub-classes share a common STN value, L_N and U_N.			

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

In the above example, R is 1 024 and the range of TSTN is 0 to 1 023, so if STN is 1 040 then TSTN = 1 040 MOD 1 024 = 16.

For Class 5 SubClass 0 and Class 5 SubClass 8, the process of validating STN and deducing MS(N) is given in Table 4.

Table 4 – Process of validating STN and deducing MS(N)

Description of the check
If LS(L_N) < LS(U_N) [so that MS(L_N) = MS(U_N)] then: if TSTN < LS(L_N) then N is rejected else if TSTN > LS(U_N) then N is rejected else N is accepted and MS(N) = MS(L_N) = MS(U_N); If LS(L_N) > LS(U_N) [so that MS(L_N) + 1 = MS(U_N)] then: if TSTN ≥ LS(L_N) then MS(N) = MS(L_N) else if TSTN ≤ LS(U_N) then MS(N) = MS(U_N) else N is rejected;

Example 1:

- Last accepted token SubClass 0 STN: 407(decimal) = 0x00000197(hexadecimal) (see Table 5 and Table 6).

Table 5 – Last accepted token example(a)

Last accepted sub-class 0 token STN value (decimal)	L_N (decimal)	U_N (decimal)	New sub class 0 token STN value (decimal)	LS(L_N)	LS(U_N)	New sub class 0 token TSTN value (decimal)	Condition Checks	MS(N)
407	24	535	408	24	535	408	LS(L_N)< LS(U_N) is TRUE.	MS(L_N) = MS(U_N) = 0

Table 6 – Last accepted token example(b)

STN	0.....23	24.....535	536.....1 023
TSTN	0.....23	24.....535	536.....1 023
Deduced MS with token acceptance or rejection status	Rejected	Accepted,MS = 0	Rejected

Example 2:

- Last accepted token SubClass 0 STN: 1023 (decimal) = 0x000003FF (hexadecimal) (see Table 7 and Table 8).

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table 7 – Last accepted token example(c)

Last Accepted sub class 0 token STN value (decimal)	L_N (decimal)	U_N (decimal)	New sub class 0 token STN value (decimal)	LS(L_N)	LS(U_N)	New sub class 0 token TSTN value (decimal)	Condition checks	MS(N)
1 023	640	1 151	1 024	640	127	0	LS(L_N)< LS(U_N) is FALSE. LS(L_N)> LS(U_N) is TRUE. TSTN ≤ LS(U_N) is TRUE.	MS(N) = MS(U_N) = 1

Table 8 – Last accepted token example(d)

STN	0.....639	640.....1 023	1 024.....1 151	1 152 and above
TSTN	0.....639	640.....1 023	0.....127	128 and above
Deduced MS with token acceptance or rejection status	Rejected	Accepted,MS = 0	Accepted, MS = 1	Rejected

If the token is accepted and STN having value N is greater than the largest STN previously accepted by the meter, the limits LN and UN are advanced according to the sub-class range table given in Table 3.

6.1.9 FunctionIndex

FunctionIndex is a 32-bit unsigned integer and is used to enforce the order of acceptance of particular tokens or groups of tokens. Whenever the TransferCredit system generates a token of a type that shall be entered into the meter (typically a token intended to change the meter's operation or tariff, as opposed to simply transferring credit), it will increase FunctionIndex by one. The meter requires every token of SubClass 8 entered to either have FunctionIndex equal to or lower than the highest FunctionIndex accepted and for SubClass 10 to have a FunctionIndex one more than the highest value already in the meter memory. Therefore, by giving two tokens Q and R of this functional group FunctionIndex values q = p+1 and r = p+2 respectively, where p is the highest FunctionIndex previously accepted for token P, the meter will not accept token R until token Q has been entered and accepted by the meter.

NOTE $q = p+1$ ensures that Q can be entered when the meter's FunctionIndex is p, and the effect of Q is to advance it from p, whereas R cannot be entered before Q while FunctionIndex is still p (as $r = p+2$ not p+1), but R can be entered after Q as $r = q+1$.

As far as the token is concerned, FunctionIndex is a partially implied value (rather than one explicitly and completely transferred within the token) that is maintained by the meter as part of the meter application process in an internal register and also maintained in the HES and used in generating and validating the MAC.

Every time a SuperToken is generated of a type that shall be entered into the meter, FunctionIndex is increased by 1.

When the meter sees a TransferCredit+Function SubClass token it will add 1 to the FunctionIndex internal register to validate the SuperToken.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

The sub-tokens within the SuperToken are controlled by the STN in the first token for anti-replay. One counter value (STN) is assumed for the complete SuperToken.

The ordering of the sub-tokens in the SuperToken is managed exclusively by the first 2 bits of the follow-on tokens indicating tokens left to arrive.

The sub-tokens within a SuperToken shall be entered in a contiguous set without other tokens or sub-tokens in the sequence.

The sub-tokens, including the first, all need to be entered in the correct order. Because the CheckDigit is calculated including preceding sub-tokens it is NOT permissible for sub-tokens 2, 3, or 4 to be entered out of order.

Validating such functional token where previous accepted token had FunctionIndex c, value (c+1) is to be used.

The first token of a type that shall be entered into the meter strictly in sequence, shall have a FunctionIndex group number of zero.

6.1.10 Relating the FunctionIndex and STN

From the STN itself FunctionIndex is computed at the meter-end. To do this, the maximum STN value of a SubClass token, already accepted by the meter, is to be saved in an array in an element corresponding to the number derived from either "FunctionIndex" or "FunctionIndex and Maximum number of array elements in the array" whenever FunctionIndex is incremented.

Relating FunctionIndex and STN is done to achieve the SubClass 8 token acceptance at the meter-end without skipping SubClass 10 tokens. Whenever any SubClass10 token passes the authentication," STN value of the token – 1" is saved in an array in memory corresponding to the element number equivalent to either {"FunctionIndex" -1} or {"Function Index MOD number of array elements"-1}. This array is referred too onwards as TokenGroupArray.

For example, for a given meter, a total of 7 tokens are issued since the meter installation and out of these 7 tokens, tokens having STN value 0x00000001,0x00000005 and 0x00000008 are of SubClass 10 then TokenGroupArray of 10 elements have values as follows:

0x00000000; 0x00000004; 0x00000007; 0x00000088; 0x00000000; 0x00000000; 0x00000000; 0x00000000; 0x00000000; 0x00000000.

Now, along with above data, the meter FunctionIndex has stored a value of 0x00000003.

So if a missed SubClass 8 token is having a TSTN value of 0x00000006 and the deduced STN is 0x00000006, then by comparing STN to STN values already stored in array elements of TokenGroupArray, true "FunctionIndex" associated with that SubClass token can be derived. If the missed token having STN 0x00000006 when entered to the meter, the meter will scan the array and check the condition for each element and condition will become TRUE for array element number 0x02. Therefore, the arrived token's FunctionIndex is 0x00000002 and this will be used to derive the TMAC of the token.

So, based on the below two cases:

- case 1: The meter's current FunctionIndex is less than or equal to the number of TokenGroupArray elements;
- case 2: The meter's current FunctionIndex is greater than the number of TokenGroupArray elements.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

By scanning from the appropriate starting array element until the arrived token's STN value is less or equal to the value stored in the array element and by applying an appropriate fomula, one can derive the value of FunctionIndex.

For case 1:

Starting from array element 0, scan and check when the arrived token's STN value is less or equal to the value in the array element. If true, then the value of FunctionIndex equals the array element number.

For case 2:

Starting from the array element "N" (where N = the current value of FunctionIndex MOD total elements in the array), scan and check when the arrived token's STN value is less or equal to the value stored in the array element. If true, then the value of FunctionIndex equals the current value of FunctionIndex plus the current array element number minus the number of elements in the array minus 1.

This is how the STN itself provides the detail of FunctionIndex to which it belongs.

For example: if the maximum STN of a token already accepted in functional group 1(SubClass 0 for unencrypted tokens or SubClass 8 "TransferCredit" and SubClass 10 "TransferCredit + Function" of encrypted tokens) is 100d then the new STN of the token type shall be higher than 100d if FunctionIndex needs to be changed.

For maximum STN value of SubClass 8 "TransferCredit" type of token already accepted with FunctionIndex 0d is 100d.

The first time SubClass 10 "TransferCredit + Function" token is being issued, the range of the token to be accepted in future is computed as follows:

a) Complete range for token:

   $L_N = M_N + 1 - 3R/8$,

   $U_N = M_N + R/8$

   M_N is the maximum value of N of the token already accepted.
   L_N is the lower limit of the STN N that is valid.

   U_N is the upper limit of the STN N that is valid.

   i.e. last accepted token STN value is 100d, L_N = 0d, U_N = 228d and M_N = 100d.

b) If the token of SubClass 10 "Transfer Credit + Function" is generated with FunctionIndex 1d and STN value (i.e. N = 101d) and when MAC and all other validation criteria of the token are found to be valid after the token validation process, (M_N – 1)d (i.e.100d) is stored at the 0th element of TokenGroupArray (which keeps record of M_N corresponding to FunctionIndex, each array element of 4 bytes) in memory and U_N (i.e.228d) is stored atthe 1st element of 4-byte UN maintaining array in memory. So, SubClass 8 "TransferCredit" tokens of type having STN value LN to MN (i.e. 0d to 100d) belongs to FunctionIndex 0d and "TransferCredit" token type having STN values belong to FunctionIndex 1d are MN+1 to UN i.e. 101d to 228d. Also in above case if UN is changing due to acceptance of "TransferCredit" token/tokens without modification in value of FunctionIndex then the meter will also update the 1st element of the array with the new UN value.

As the next step the vending system shall generate another TransferCredit token with FunctionIndex 1d only and STN value from the range 102d to 228d. (Say TranferCredit token having STN value 101 with incremented FunctionIndex with value 1).

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

The number of elements in TokenGroupArray in memory, which keeps record of "relating FunctionIndex and STN", should be kept considering the following three factors:

- How frequently does the tariff change in the particular utility market?;
- What is the expected life of the meter?;
- How many Class 5 SubClass 10 "TranferCredit+Function" tokens are expected to be issued due to tariff changes?.

6.1.11 SingleTokenPayload

The SingleTokenPayload is a 61-bit value made up as follows:

a) The 32-bit TMAC which is truncated from LS part of 128 bit MAC and placed in LS 32 bits of the 61-bit Token data (see 6.2);

b) The remaning 29-bit value is padded out with 3 bits having value 000b in the MS position to create a 32-bit number.

The content and meaning of the MS 29 bits of 61 bits value of SingleTokenPayload vary according to the token SubClass (see 6.2).

For example: Class 5 SubClass 0 (see 6.2.4.1), Class 5 SubClass 8 (see 6.2.5.1) and Class 5 SubClass 9 (see 6.2.5.2) are having SingleTokenPayload with one data block.

Refer to Figure 9 and 6.1.15 to understand TMAC derivation for SingleTokenPayload.

6.1.12 SuperTokenPayload

The SuperTokenPayload shall be a variable length of bits of the APDU, that is dependent on the number of data blocks that are considered together to make a token.

The payload of SuperTokenPayload varies depending on the number of data blocks in the SuperToken, i.e. 122 bits (61 bits + 61 bits), 183bits (61 bits + 61 bits + 61 bits), 244 bits (61 bits + 61 bits + 61 bits + 61 bits) for 2, 3 or 4 data blocks respectively. Though SuperTokenPayload size is mentioned in bits above for a discrete number of data blocks, each data block related 61 bits are stored in 8 bytes storage space by keeping each block 3 MS bits as 000b.

For example: Class 5 SubClass 10 having SuperTokenPayload (see 6.2.5.3).

Refer to Figure 10 and 6.1.16 to understand TMAC derivation for SuperTokenPayload.

6.1.13 MessageAuthenticationCode (MAC) and TruncatedMAC (TMAC)

6.1.13.1 MAC calculation

The 128-bit MAC calculation is performed over the following data elements for SingleTokenPayload in accordance with Figure 5.

In Figure 5, Generic Block 0 is shown as full 64 bits, but only the MS 32 bits are taken in AAD payload construction as per token SubClass. This is also described in 6.1.14.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

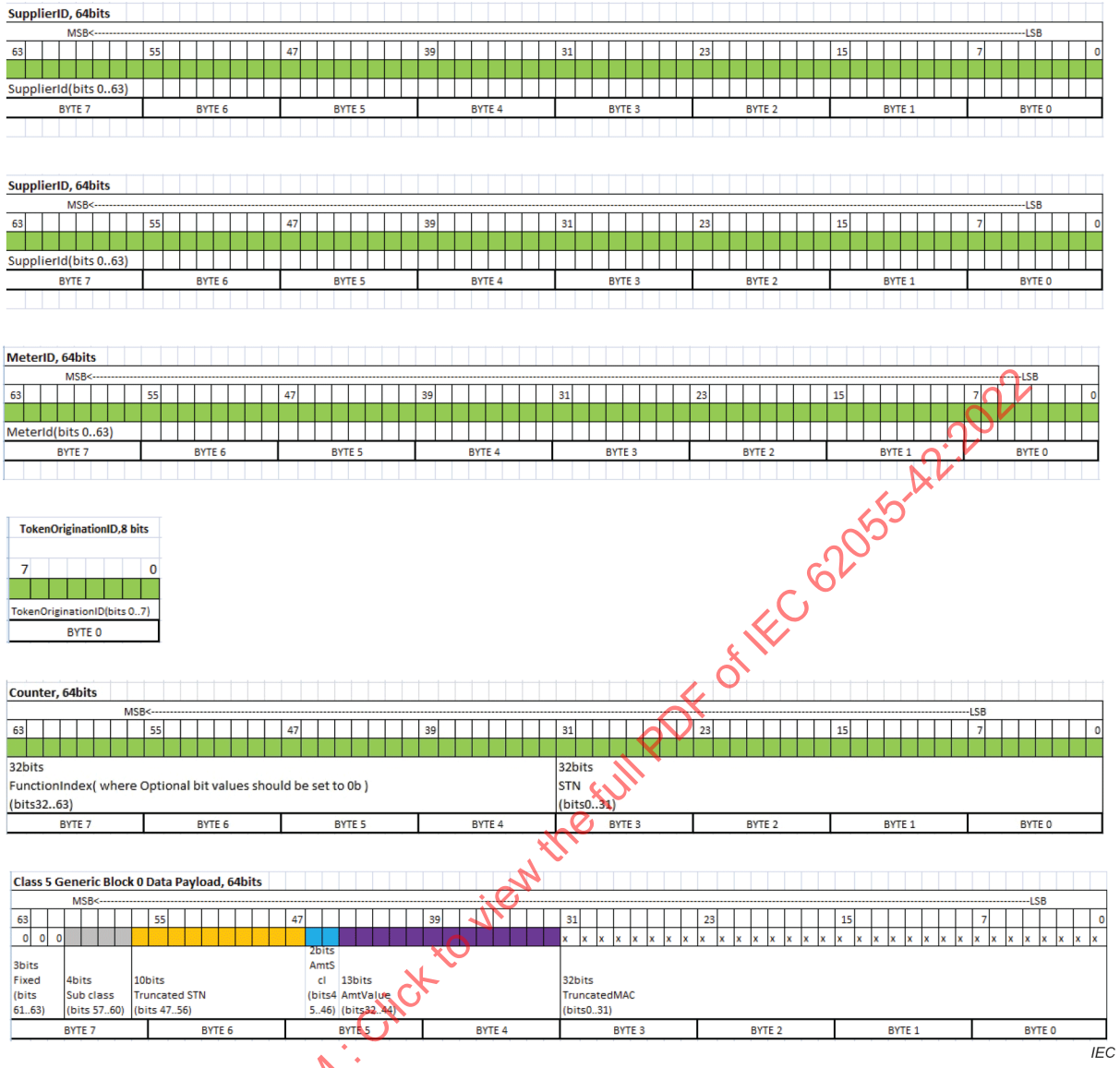


Figure 5 – Generic data elements for AAD payload construction for SingleTokenPayload

The 128-bit MAC calculation is performed over the following data elements for SuperTokenPayload in accordance with Figure 6.

In Figure 6, Generic Block 0 is shown fully, but only the MS 32 bits are taken in AAD payload construction as per token SubClass. This is also described in 6.1.14.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

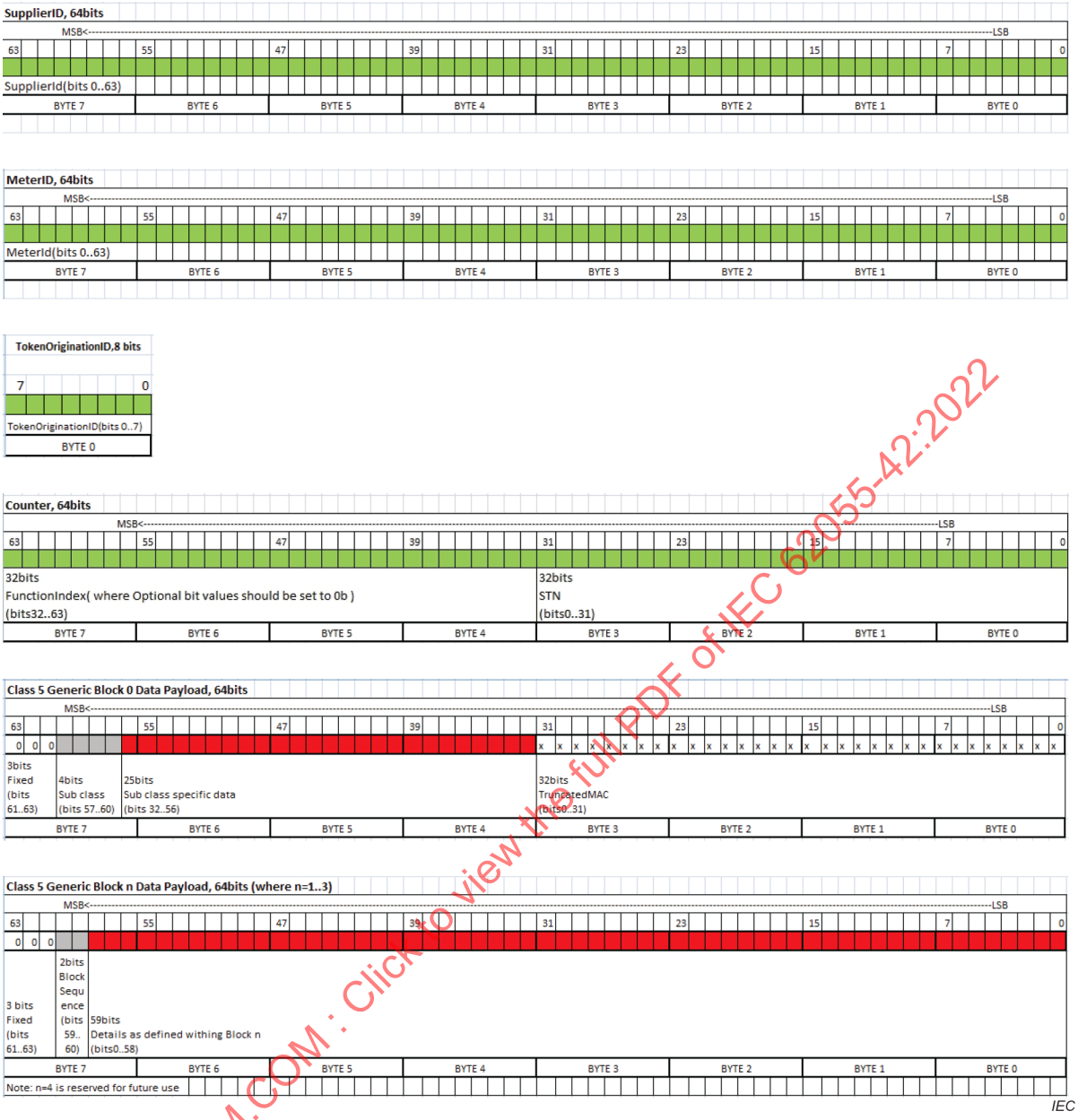


Figure 6 – Generic data elements for AAD payload construction for SuperTokenPayload

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Figure 7 gives the information on how to prepare the IV payload using InvocationField and SuppliedID data elements.

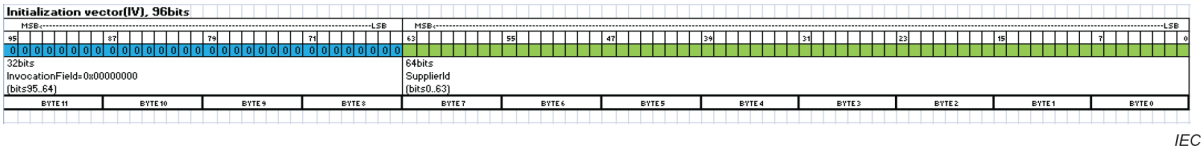


Figure 7 – InitializationVector (IV) construction

6.1.13.2 GMAC construction

Figure 8 shows data elements taken as input to the GMAC Engine to derive the MAC value. The 32 LS bits of this MAC value is used as TMAC in the APDU construction.

The GMAC algorithm, based on the use of AES-128, is used for the calculation of the MAC as specified in NIST SP 800-38D: 2007.

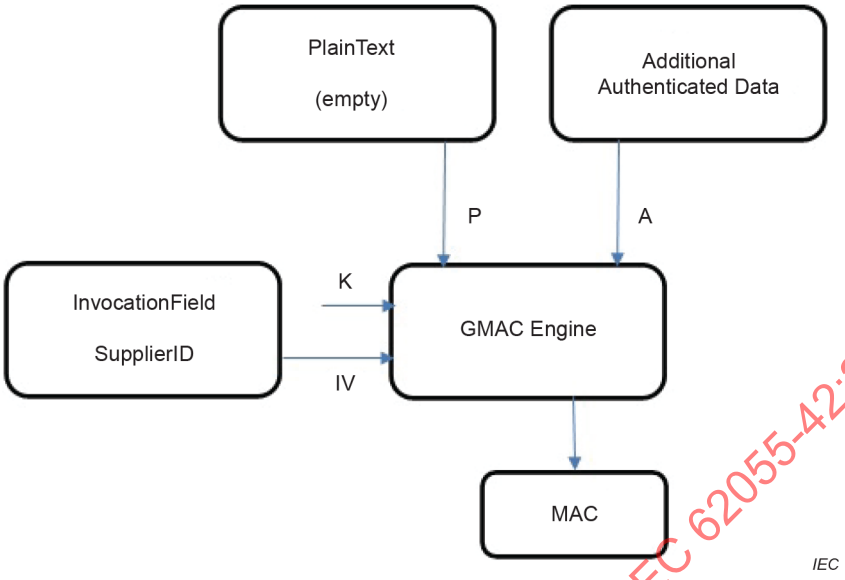


Figure 8 – GMAC construction

6.1.14 AdditionalAuthenticationData (AAD)

AAD shall be made up of the following concatenation:

AAD = MessageIdentifier || 32 MS bits of generic block 0 data payload || 64 bits of generic block 1 payload (optional) || 64 bits of generic block 2 payload (optional) || 64 bits of generic block 3 payload (optional).

NOTE 1 The number of block payloads in AAD depends on the sub-class and the data to be sent in the sub-class.

NOTE 2 Sub-class detail and block sequence numbers are not included in AAD.

6.1.15 SingleTokenPayload AAD preparation, TMAC derivation and APDU preparation

This subclause describes how to prepare AAD for the SingleTokenPayload based on data elements for SubClass 8 token, derivation of TMAC and thereby updating of the complete APDU for the SubClass 8 token.

Figure 9 shows a TMAC derivation example for SubClass 8 token considering the approach given in Figure 8.

Below are data elements either in basic value form or in hexadecimal value for:

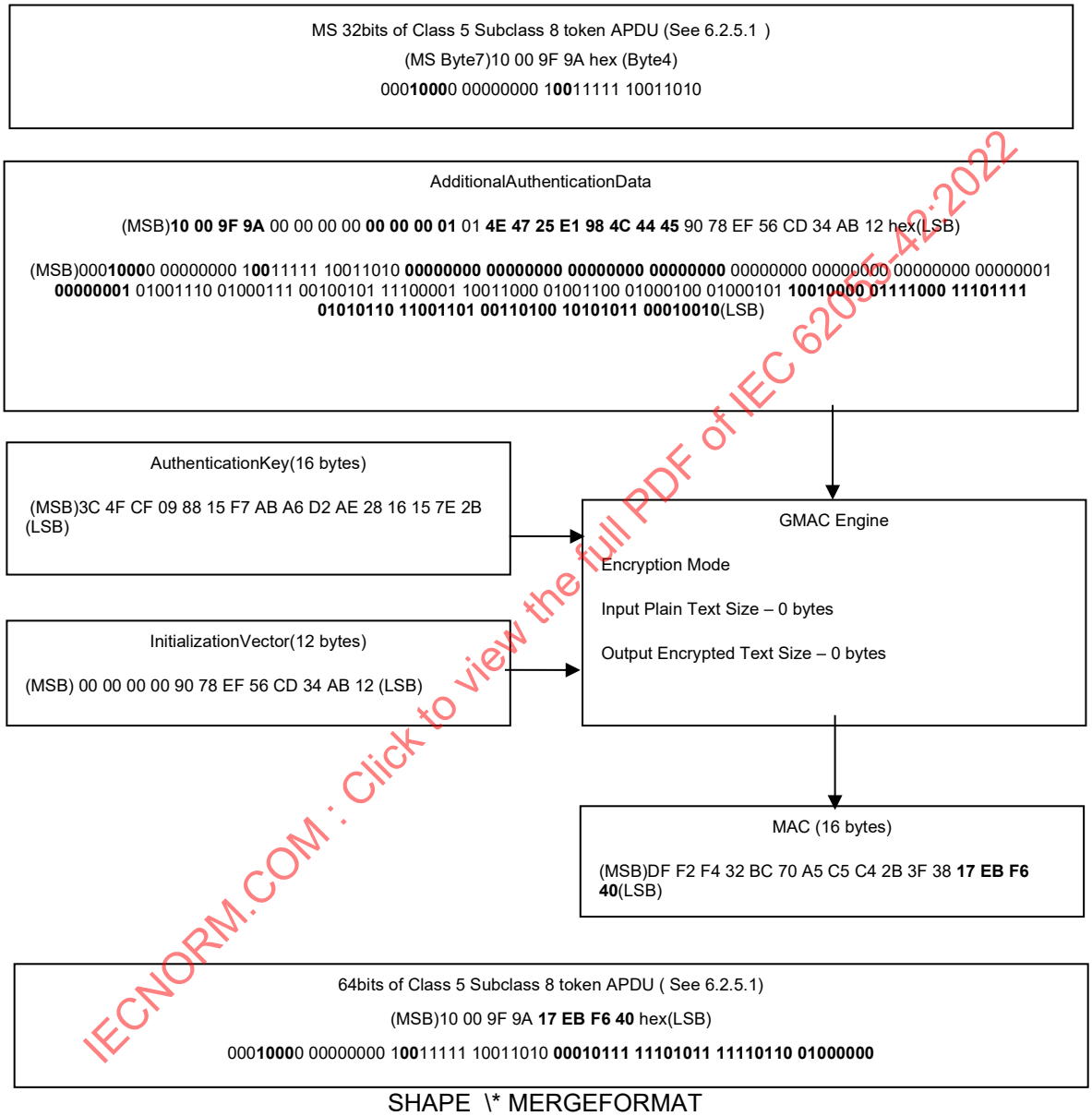
- SupplierID: 90 78 EF 56 CD 34 AB 12;
- MeterID: 45 44 4C 98 E1 25 47 4E;
- TokenOriginationID: 0x01;
- STN: 00 00 00 01;
- FunctionIndex: 00 00 00 00;
- Amount to transfer: 8090d.

Derived IV and MessageIdentifier instances are as follows:

- InitializationVector Buffer: 12 AB 34 CD 56 EF 78 90 00 00 00 00;

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

- MessageIdentifier Buffer: **12 AB 34 CD 56 EF 78 90** 45 44 4C 98 E1 25 47 4E **01** 01 00 00 00 **00 00 00 00**;
- 32 MS bits of SingleTokenPayload: Block1(Byte7)10 00 9F 9A (Byte4);
- AdditionalAuthenticationDataBuffer: **12 AB 34 CD 56 EF 78 90** 45 44 4C 98 E1 25 47 4E **01** 01 00 00 00 **00 00 00 00** 9A 9F 00 10;
- AuthenticationKey: 3C 4F CF 09 88 15 F7 AB A6 D2 AE 28 16 15 7E 2B.



SHAPE * MERGEFORMAT

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Figure 9 – Class 5 SubClass 8 TMAC derivation and full APDU preparation example

6.1.16 SuperTokenPayload AAD preparation, TMAC derivation and APDU preparation

This subclause describes how to prepare the AAD for the SuperTokenPayload based on data elements for the SubClass 10 token, derivation of TMAC and thereby updating of the comlete APDU for the SubClass 10 token.

Figure 10 shows a TMAC derivation example for a SubClass 10 token considering the approach shown in Figure 8.

Below are data elements either in basic value form or in hexadecimal value form (as stored in memory in little endian format):

- SupplierID: 90 78 EF 56 CD 34 AB 12;
- MeterID: 45 44 4C 98 E1 25 47 4E;
- TokenOriginationID: 0x01;
- STN: 00 00 00 02;
- FunctionIndex: 00 00 00 01;
- Amount to transfer: 8090d;
- Slab tariff with 8 slabs: 355, 600, 900, 1200, 1600, 2000, 2600;
- ActivationMonth: Feb 2019.

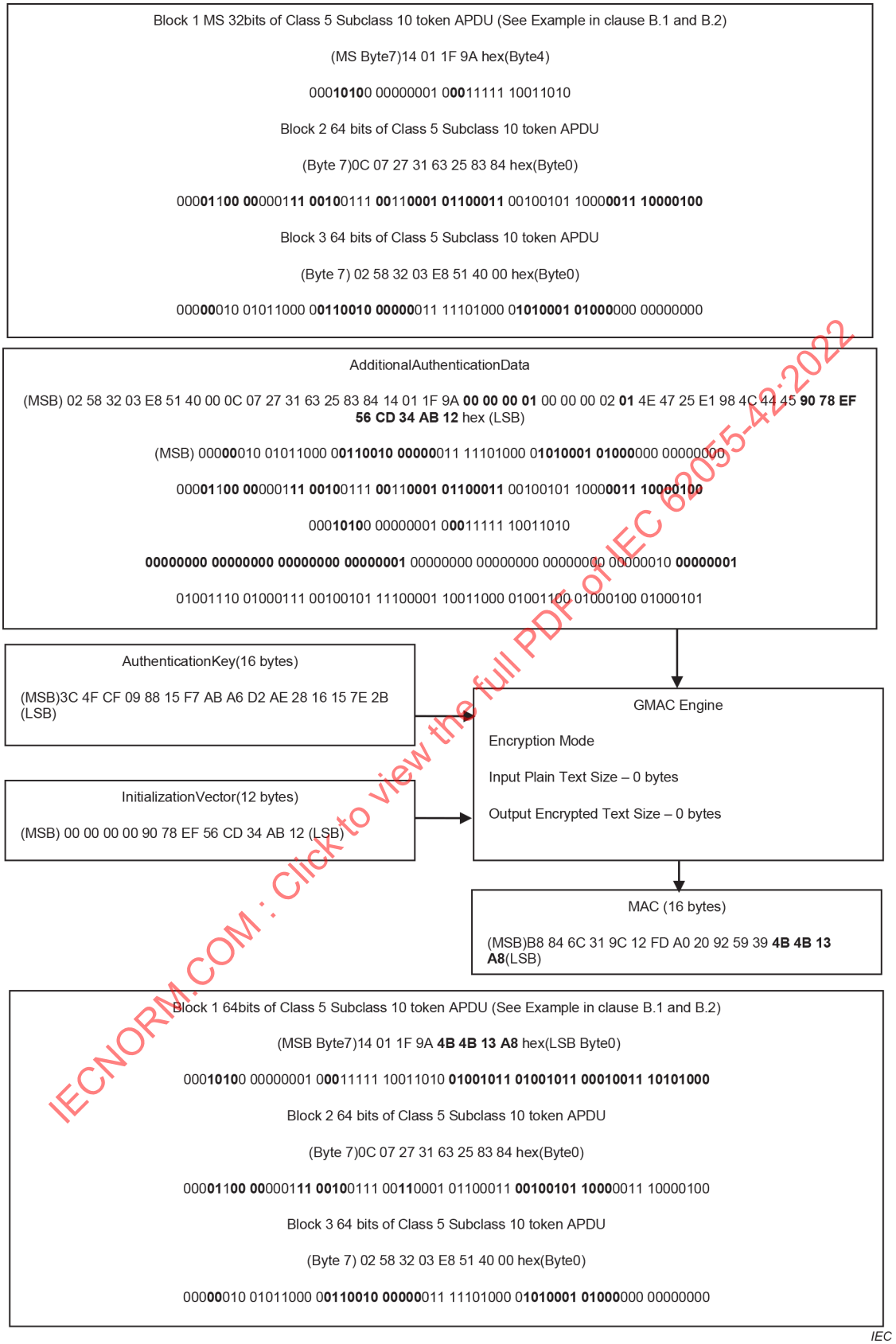
Derived IV and MessageIdentifier instances are as below:

- InitializationVectorBuffer: 12 AB 34 CD 56 EF 78 90 00 00 00 00;
- MessageIdentifierBuffer: **12 AB 34 CD 56 EF 78 90** 45 44 4C 98 E1 25 47 4E **01** 02 00 00 00 **01 00 00 00**;
- SuperTokenPayload:
  - Block 1 (Byte7)14 01 1F 9A(Byte4);
  - Block 2 (Byte7)0C 07 27 31 63 25 83 84(Byte0);
  - Block 3 (Byte7)02 58 32 03 E8 51 40 00(Byte0).

AdditionalAuthenticationDataBuffer:

- **12 AB 34 CD 56 EF 78 90** 45 44 4C 98 E1 25 47 4E **01** 02 00 00 00 **01 00 00 00** 9A 1F 01 14 84 83 25 63 31 27 07 0C 00 40 51 E8 03 32 58 02;
- AuthenticationKey: 3C 4F CF 09 88 15 F7 AB A6 D2 AE 28 16 15 7E 2B.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022



IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Figure 10 – Class 5 SubClass 10 TMAC derivation and full APDU preparation example

6.1.17 Offset

In the calculation of the Class 4 tokens and above, various offsets are needed in order to access the correct token domain when assembling the APDU and transforming to the TCDU. Table 9 defines these offsets and their values together with the minimum and maximum values of the several classes of tokens.

The values are given in both decimal and hexadecimal. The decimal values can be used to determine by inspection into which class a decimal token falls, but it may be convenient to add or subtract an offset using binary arithmetic and this document does not prescribe whether binary or decimal arithmetic is used.

Table 9 – Numeric constants and their purpose

Constant	Value – decimal	Value – hexadecimal	Purpose
K_Class_0_1_2_3_MIN	00 000 000 000 000 000 000	0x0 0000 0000 0000 0000	MIN Class 0, 1, 2, 3 token value
K_Class_0_1_ 2_3_MAX	73 786 976 294 838 206 463	0x3 FFFF FFFF FFFF FFFF	MAX Class 0, 1, 2, 3 token value
K_Class _4_MIN	73 786 976 294 838 206 470	0x4 0000 0000 0000 0006	MIN Class 4 token value
K_Class_4_MAX	73 941 569 907 863 060 479	0x4 0225 3A12 6CE3 FFFF	MAX Class 4 token value
K_Class 5_OFFSET (19 digit)	7 394 156 990 786 306 048	0x669D 529B 714A 0000	Constant to add on to Class 5 token
K_Class_5_MIN	73 941 569 907 863 060 480	0x4 0225 3A12 6CE4 0000	MIN Class 5 token value
K_Class_5_MAX	96 999 999 999 999 999 999	0x5 4225 3A12 6CE3 FFFF	MAX Class 5 token value
K_Class_6_MIN	97 000 000 000 000 000 000	0x5 4225 3A12 6CE4 0000	Additional classes yet to be specified start here
K_Class_X_MAX	99 999 999 999 999 999 999	0x5 6BC7 5E2D 630F FFFF	End of token domain (highest possible 20-digit decimal number)

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.2 Tokens

6.2.1 Token definition and format

The token payload comprises the 29-bit value plus the 32-bit TMAC in the APDU (see 6.1.1, 6.1.11 and 6.1.12) to make up a 61-bit binary SingleTokenPayload comprising a number of fields of smaller data elements, specified in this subclause. The token domains specified in this document shall not overlap with the framework provided by the existing STS standard IEC 62055-41:2018 and indeed shall provide a new domain of operation outside that of STS token classes 0, 1, 2, and 3. There are a number of departures from the currently specified IEC 62055-41:2018 that are required for the Class 5 token, which are mainly the ability to have an unencrypted token, an encrypted token and the ability to have a built-in TMAC for token authentication.

The definition and format of information for the tokens are given in Table 10.

Table 10 – Token definition and format

Name of data element	e.g. Amount, MAC, Class, SubClass, etc.
Data format	e.g. Binary, Decimal, etc.
Number of bits	e.g. 15 bits, etc.
Range of values	e.g. 0-9, 1,3, etc.

Tokens are defined into classes depending on their function as follows:

- Class 0 – exclusively defined in IEC 62055-41:2018;
- Class 1 – exclusively defined in IEC 62055-41:2018;
- Class 2 – exclusively defined in IEC 62055-41:2018;
- Class 3 – exclusively defined in IEC 62055-41:2018;
- Class 4 – exclusively defined in this document;
- Class 5 – exclusively defined in this document.

6.2.2 Class 4: RESERVED FOR FUTURE ASSIGNMENT

This token class range is reserved for future allocation by IEC TC 13 WG 15 and shall not be used for any other purpose.

6.2.3 Class 5 tokens

6.2.3.1 General

The token range of Class 5 is divided into 16 types of token identified by use of the SubClass field. These 16 types are split into 2 categories; unencrypted tokens and encrypted tokens. Values of SubClass from 0 to 7 shall be used for unencrypted tokens and values of SubClass from 8 to 15 shall be used for encrypted tokens as given in Table 11.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table 11 – Class 5 SubClass assignment

Token SubClass	Token Class 5	
	NON-ENCRYPTED	ENCRYPTED
0	TransferCredit token	Not permitted
1	Reserved for SpecialFunction	Not permitted
2	Reserved	Not permitted
3	Reserved	Not permitted
4	Reserved	Not permitted
5	Reserved	Not permitted
6	Reserved	Not permitted
7	Reserved	Not permitted
8	Not permitted	TransferCredit token
9	Not permitted	SpecialFunction token
10	Not permitted	ExtendedTransferCredit token
11	Not permitted	MeterGenerated token
12	Not permitted	Reserved
13	Not permitted	Reserved
14	Not permitted	Reserved
15	Not permitted	Reserved

Table 12 shows the decimal domain ranges of the APDU payload for the 61-bit binary domain as assigned for sub-classes within Class 5 (before encryption), and Table 13 shows the token APDU converted into the decimal domain with the Class 5 offset added to the 61-bit APDU decimal equivalent (before the check digit is added to form the TCDU).

Thus a device can determine the SubClass by inspection of the decimal value without decryption, the MS 4 bits (bits 57 to 60) of the 61-bit token block, containing the SubClass, shall not be encrypted. If the token consists of several blocks then the sequence number in the MS 2 bits (bits 59 and 60) of the following blocks (for blocks other than 1st block) shall not be encrypted.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table 12 –SubClass-wise boundaries for Class 5 APDU before encryption

SubClass (binary)	Minimum payload value (decimal)	Minimum payload value (hexadecimal)	Maximum payload value (decimal)	Maximum payload value (hexadecimal)
0000	000000000000000000	0000000000000000	144115188075855871	01FFFFFFFFFFFFF
0001	144115188075855872	0200000000000000	288230376151711743	03FFFFFFFFFFFFF
0010	288230376151711744	0400000000000000	432345564227567615	05FFFFFFFFFFFFF
0011	432345564227567616	0600000000000000	576460752303423487	07FFFFFFFFFFFFF
0100	576460752303423488	0800000000000000	720575940379279359	09FFFFFFFFFFFFF
0101	720575940379279360	0A00000000000000	864691128455135231	0BFFFFFFFFFFFFF
0110	864691128455135232	0C00000000000000	1008806316530991103	0DFFFFFFFFFFFFF
0111	1008806316530991104	0E00000000000000	1152921504606846975	0FFFFFFFFFFFFFF
1000	1152921504606846976	1000000000000000	1297036692682702847	11FFFFFFFFFFFFF
1001	1297036692682702848	1200000000000000	1441151880758558719	13FFFFFFFFFFFFF
1010	1441151880758558720	1400000000000000	1585267068834414591	15FFFFFFFFFFFFF
1011	1585267068834414592	1600000000000000	1729382256910270463	17FFFFFFFFFFFFF
1100	1729382256910270464	1800000000000000	1873497444986126335	19FFFFFFFFFFFFF
1101	1873497444986126336	1A00000000000000	2017612633061982207	1BFFFFFFFFFFFFF
1110	2017612633061982208	1C00000000000000	2161727821137838079	1DFFFFFFFFFFFFF
1111	2161727821137838080	1E00000000000000	2305843009213693951	1FFFFFFFFFFFFFF

Table 13 – SubClass-wise boundaries for Class 5 tokens, TCDU after encryption (if applicable) and adding offset (without CheckDigit)

SubClass – binary	minimum value – decimal	minimum value – hexadecimal	maximum value – decimal	maximum value – hexadecimal
0000	7394156990786306048	669D529B714A0000	7538272178862161919	689D529B7149FFFF
0001	7538272178862161920	689D529B714A0000	7682387366938017791	6A9D529B7149FFFF
0010	7682387366938017792	6A9D529B714A0000	7826502555013873663	6C9D529B7149FFFF
0011	7826502555013873664	6C9D529B714A0000	7970617743089729535	6E9D529B7149FFFF
0100	7970617743089729536	6E9D529B714A0000	8114732931165585407	709D529B7149FFFF
0101	8114732931165585408	709D529B714A0000	8258848119241441279	729D529B7149FFFF
0110	8258848119241441280	729D529B714A0000	8402963307317297151	749D529B7149FFFF
0111	8402963307317297152	749D529B714A0000	8547078495393153023	769D529B7149FFFF
1000	8547078495393153024	769D529B714A0000	8691193683469008895	789D529B7149FFFF
1001	8691193683469008896	789D529B714A0000	8835308871544864767	7A9D529B7149FFFF
1010 [1)]	8835308871544864768	7A9D529B714A0000	8979424059620720639	7C9D529B7149FFFF
1011 [1)]	8979424059620720640	7C9D529B714A0000	9123539247696576511	7E9D529B7149FFFF
1100	9123539247696576512	7E9D529B714A0000	9267654435772432383	809D529B7149FFFF
1101	9267654435772432384	809D529B714A0000	9411769623848288255	829D529B7149FFFF
1110	9411769623848288256	829D529B714A0000	9555884811924144127	849D529B7149FFFF
1111	9555884811924144128	849D529B714A0000	9699999999999999999	869D529B7149FFFF

NOTE 1 SubClass 10 and SubClass 11 have more than 1 data block of APDU and so have more than 20 digits of TCDU, thus above table range is applicable for data block 1 TCDU only.

NOTE 2 The values are before adding the CheckDigit.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table 14 shows the boundaries for TCDUs in all 16 sub-classes within Class 5. Because of the nature of the check digit it will be likely that the decimal representations of the true minimum and maximum tokens will not end with 0 and 9 respectively. This is because for every 19-digit token payload there will only be one check digit possible, and so only one in ten 20-digit TCDUs will be legitimate. However, the entire decimal space in the 20-digit domain is reserved for the appropriate sub-classes regardless. Further, it SHALL NOT be permitted to use token values

within the boundaries that do not have a valid check digit for any other purpose, including manufacturer-specific purposes.

Table 14 – Class 5 SubClass boundaries for TCDU (reserved space)

SubClass – binary	Actual TCDU minimum	Actual TCDU maximum
0000	73941569907863060480	75382721788621619199
0001	75382721788621619200	76823873669380177919
0010	76823873669380177920	78265025550138736639
0011	78265025550138736640	79706177430897295359
0100	79706177430897295360	81147329311655854079
0101	81147329311655854080	82588481192414412799
0110	82588481192414412800	84029633073172971519
0111	84029633073172971520	85470784953931530239
1000	85470784953931530240	86911936834690088959
1001	86911936834690088960	88353088715448647679
1010 [1)]	88353088715448647680	89794240596207206399
1011 [1)]	89794240596207206400	91235392476965765119
1100	91235392476965765120	92676544357724323839
1101	92676544357724323840	94117696238482882559
1110	94117696238482882560	95558848119241441279
1111	95558848119241441280	96999999999999999999
NOTE Subclass 10 and Subclass 11 have more than 1 data block of APDU and so have more than 20 digits of TCDU (40 or 60 or 80 digits TCDU), thus range is applicable for the first 20 digits of TCDU only.		

6.2.3.2 Class 5 token SubClass related FunctionalClass and associated use case

Each SubClass token acceptance at the meter end has a certain behaviour/functional approach based on its practical use case as shown in Table 15. This behaviour/functional approach is described as FunctionalClass number in this document.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table 15 – SubClass related FunctionalClass and associated use cases

SubClass	FunctionalClass	Behaviour	Use case
0	1	Tokens need not be entered into the meter in the exact sequence in which they were issued, or at all – there is considerable latitude (but not complete freedom) in the order of entry.	If two TransferCredit tokens are issued to a consumer then they can be entered into the meter in any order.
8	1	Tokens need not be entered into the meter in the exact sequence in which they were issued, or at all – there is considerable latitude (but not complete freedom) in the order of entry.	If two TransferCredit tokens are issued to a consumer then they can be entered into the meter in any order.
9	3	Tokens shall be entered into the meter in the sequence in which they were issued, but some may be omitted without affecting later transactions. Such omitted transactions cannot then subsequently be entered. A separate STN is maintained for SpecialFunction tokens compared to "TranferCredit" and "Transfer Credit+Function" tokens although they use the same format. Similaraly a SpecialFunction token will have its own TSTN compared to that of "Tranfer Credit" and "Transfer Credit+Function" tokens. All SpecialFunction tokens have to be entered into the meter in order if one wants all the tokens to be accepted and once aged out will be invalid. Therefore tokens created in the future can be entered and will be accepted; however, older unused tokens will not be accepted by the meter.	A SpecialFunctionToken is superseded by another SpecialFunctionToken issued subsequently. The earlier token will not be accepted after the later one.
10	1	Tokens shall be entered into the meter strictly in the sequence in which they were issued without omissions.	The consumer is issued a "TransferCredit + Function" token. This shall be entered before any subsequently purchased TransferCredit/ "TransferCredit + Function" token. If above rule is fulfilled then it will be accepted.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Tokens of SubClass 8 and SubClass 10 both have amount detail and therefore there should be a common STN maintained for two types of sub-classes and parameters L_N, U_N and M_N should be maintained as common for these two SubClass type of tokens. So STN, L_n, U_n and M_n are to be maintained Functional SubClass-wise at the meter-end and there should be a mapping of SubClass and FunctionalClass number at the meter-end.

6.2.4 Class 5: Unencrypted tokens

6.2.4.1 SubClass 0: TransferCredit token

Table 16 gives the structure for a SubClass 0 TransferCredit token.

Table 16 – SubClass 0: TransferCredit token

SubClass	TSTN	AMTConfig	AMT	TMAC
Binary	Binary	Binary	Binary	Binary
4 bits	10 bits	2 bits	13 bits	32 bits
0 = Unencrypted TransferCredit See 6.2.3	Truncated 32-bit number	0d – 3d See 6.3	0d – 8191d See 6.3	Truncated 128-bit GMAC

AMTConfig determines the interpretation of AMT as follows:

- 0d – Means AMT is 0d to 8191d in steps of 1;
- 1d – Means AMT is 0d to 819100d in steps of 100;
- 2d – Means AMT is 0d to 81910000d in steps of 10000;
- 3d – Means AMT is 0d to 8191000000d in steps of 1000000.

6.2.4.2 SubClass 1: SpecialFunction tokens

This subclause specifies the token format for SpecialFunction tokens that are assigned to SubClass 1 within Class 5.

The encoding structure is to be defined by IEC TC 13/WG 15 and may not be used for any other purpose.

6.2.4.3 SubClass 2: RESERVED

This SubClass is reserved for future assignment by IEC TC 13/WG 15 and may not be used for any other purpose.

6.2.4.4 SubClass 3: RESERVED

This SubClass is reserved for future assignment by IEC TC 13/WG 15 and may not be used for any other purpose.

6.2.4.5 SubClass 4: RESERVED

This SubClass is reserved for future assignment by IEC TC 13/WG 15 and may not be used for any other purpose.

6.2.4.6 SubClass 5: RESERVED

This SubClass is reserved for future assignment by IEC TC 13/WG 15 and may not be used for any other purpose.

6.2.4.7 SubClass 6: RESERVED

This SubClass is reserved for future assignment by IEC TC 13/WG 15 and may not be used for any other purpose.

6.2.4.8 SubClass 7: RESERVED

This SubClass is reserved for future assignment by IEC TC 13/WG 15 and may not be used for any other purpose.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.2.5 Class 5: Encrypted tokens

6.2.5.1 SubClass 8: TransferCredit token

Table 17 gives the structure for a SubClass 8 TransferCredit token.

Table 17 – SubClass 8: TransferCredit token

SubClass	TSTN	AMTConfig	AMT	TMAC
Binary	Binary	Binary	Binary	Binary
4 bits	10 bits	2 bits	13 bits	32 bits
8 = 1000b Encrypted TransferCredit token See 6.2.3	Truncated 32-bit number	0d – 3d	0d – 8191d	Truncated 128-bit GMAC

AMTConfig determines the interpretation of AMT as follows:

- 0d – Means AMT is 0d to 8191d in steps of 1;
- 1d – Means AMT is 0d to 819100d in steps of 100;
- 2d – Means AMT is 0d to 81910000d in steps of 10000;
- 3d – Means AMT is 0d to 8191000000d in steps of 1000000.

6.2.5.2 SubClass 9: SpecialFunction tokens

6.2.5.2.1 General

This subclause specifies the token format for SpecialFunction tokens that are assigned to SubClass 9 within Class 5.

The encoding structure for a SpecialFunction token is given in Table 18.

The data fields SubClass, TSTN, and TMAC in this token shall be according to the definitions in 6.2.3, 6.1.7 and 6.1.13.

Table 18 – Class 5, SubClass 9: SpecialFunction token

SubClass	TSTN	Data		TMAC
		ServiceType	RES	
4 bits	10 bits	5 bits	10 bits	32 bits
9 = 1001b – Engineering See 6.2.3	Truncated 32-bit number	See Table 19.	Reserved	Truncated 128-bit MAC
NOTE 1 The check digit is not shown in this table as it applies to the decimal representation of the token when converted to TCDU. NOTE 2 The 4-bit SubClass part of this token is not encrypted. The remaining 57 bits are encrypted. NOTE 3 The computation of the MAC does not include the FunctionIndex for this SubClass value.				

The supported service types are given in Table 19.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table 19 – Service types

ServiceType	Description	Remarks
0	Arm load control switch (LCS)	See 6.2.5.2.3
1	Reset PIN	See 6.2.5.2.4.
2	Engineering mode	See 6.2.5.2.5.
3	Disconnect LCS	See 6.2.5.2.6
4	Reset tamper and arm LCS	See 6.2.5.2.7.
5	Connect LCS (if LCS is in armed state)	See 6.2.5.2.8.
6	Reset allowed overload count	See 6.2.5.2.9
7	Disconnect LCS and generate balance settlement token	See 6.2.5.2.10
8 to 31	Reserved	

6.2.5.2.2 ServiceType

This data field is a simple enumeration designed to initiate a predefined function or behavior, described in Table 19. The support for these additional service tokens shall be optional within the wider standard but may be mandatory within companion specifications. If a particular token type is not supported, then the meter shall return the verification error 8 – FunctionError.

6.2.5.2.3 Arm load control switch (LCS)

This token instructs the meter to change the state of the LCS from Disconnected to Armed. From the state of Armed the consumer can then take some additional action, not defined in this document, to reconnect the LCS.

NOTE This is intended for use for example in a situation when a meter has been given a command over the WAN to disconnect LCS and the WAN has subsequently become unavailable, leaving the consumer without supply. The LCS can be reconnected using this token with or without WAN communications to the meter.

6.2.5.2.4 Reset PIN

This token allows a consumer to reset their meter's PIN to a default value (for example, 0000).

NOTE The consumer is then able to choose another PIN; this is outside the scope of this document.

PIN management details shall be agreed between the meter vendor and the utility.

6.2.5.2.5 Engineering mode

This token allows the meter to be put into an engineering mode for a limited period, to perform functions that are not normally exposed to the consumer, for example by enabling installation operations and additional displays.

6.2.5.2.6 Disconnect LCS

This token allows the utility to disconnect the LCS, for example as a penalty for tamper.

6.2.5.2.7 Reset tamper and arm LCS

This token allows the utility to reset tamper flags in the meter and set the LCS to the armed state. The consumer can then take some action, not defined in this document, to reconnect the supply.

The details of consumer action shall be agreed between the meter vendor and the utility.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.2.5.2.8 Connect LCS

This token connects the supply using the LCS. It has no effect unless the LCS is already in the armed state.

NOTE Its intended use is in emergency situations when normal consumer interaction is not possible, for example when the necessary consumer input device such as a keypad is faulty. In such circumstances, the token can be sent via the WAN to reconnect the supply without a site visit.

6.2.5.2.9 Reset allowed overload count

The meter may respond to an overload condition by using the LCS to disconnect the supply, following which the consumer can take some action to reconnect the supply. The number of such disconnect-reconnect cycles may be limited (for example to a daily maximum) to ensure the life of the meter is not impaired.

This token allows the utility to override such a limit by resetting the count of disconnect-reconnect cycles so that the consumer can again reconnect supply.

6.2.5.2.10 Disconnect LCS and generate balance settlement token

This token may be used on change of tenancy. It allows the utility to cause the meter to disconnect the LCS and generate a token encoding and authenticating the account balance.

The resulting meter-generated token, due to this special function token, is of Class 5 SubClass 11 and is described in 6.2.5.4. It allows the customer to securely report the closing account balance to the utility to facilitate repayment of the credit balance in the meter.

6.2.5.3 SubClass 10: Extended TransferCredit tokens

The generic structure for the Class 5 SubClass 10 token is given in Table 20.

Table 20 – Block 1 of TransferCredit + Function token

SubClass	TSTN	Data		TMAC
		AMTConfig	AMT	
4 bits	10 bits	2 bits	13 bits	32 bits
10 = 1010b See 6.2.3	See NOTE 2			See NOTE 2
NOTE 1 The TSTN, AMTConfig, AMT, TMAC field valid values and their interpretations are the same as those of the corresponding fields of the token of Class 5 SubClass 8 (see 6.2.5.1). NOTE 2 The TMAC field is truncated from 128 bits of GMAC calculated over all participating blocks. The method of calculating the 128-bit GMAC is detailed in 6.1.13, 6.1.15 and 6.1.16.Subsequent tokens in the token group will have a different structure and shall be entered into the meter in sequential order so that the entire SuperToken may be reassembled by the meter application process.				

For Class 5 SubClass 10 tokens, the MS 4 bits of the first block (the SubClass) shall be transmitted unencrypted and the MS bit shall be used to determine whether all blocks of the token are encrypted or unencrypted.

The order of the tokens in the group is labelled in the structure of the subsequent tokens by means of a counter in the first 2 bits of the token. The counter will designate how many further tokens there are to follow (see Table 21) with the final token in the group or SuperToken having BlockSequence = 0 as per Table 22.

The CheckDigit in the first token applies to the first 19 digits of the first token in the group only, whereas the subsequent tokens of the group have check digits that are calculated as follows.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

For the Nth token T_N (N > 1), prepend the check digit C_{N-1} of the preceding token T_{N-1} to the 19 digits of T_N and calculate the check digit of the resulting 20 digits C_{N-1} || T_N.

Example on how to prepend data to calculate CheckDigit of the second block of the 40 digit TCDU as follows:

- 40 Digit TCDU: 8889793723820927018101660992186693955792;
- For the TCDU's first 20 digits related CheckDigit is to be derived from the first 19 digits (i.e. "8889793723820927018") using Verhoeff algorithm and CheckDigit value as "1";
- For the TCDU's second block related CheckDigit is to be derived from "1" (i.e. 20th digit), prepended to the second block of 19 digits ("0166099218669395579") and overall 20 digits ("10166099218669395579") using the Verhoeff algorithm and CheckDigit value as "2".

The result of the calculation is the CheckDigit C_N to be appended to T_N.

Table 21 – Block 2 to *N-1* of *N* (*N* > 2) TransferCredit + Function token

BlockSequence	Data
2 bits	59 bits
01b – means one more block follows 10b – means two more blocks follow 11b – Reserved See B.1.2	Application specific semantic meaning to be covered in Companion Specifications

Table 22 – Last block TransferCredit + Function token

BlockSequence	Data
2 bits	59 bits
= 00b See B.1.2	Application specific semantic meaning to be covered in Companion Specifications

For an example of how the sequential tokens are constructed, see Annex B.

6.2.5.4 SubClass 11: Encrypted meter generated token

6.2.5.4.1 General

This token is generated by the meter for the purpose of providing a secure message to a POS or HES in order to provide the consumer with a refund for unused credit.

Block 1: Meter generated token for ReturnedMeterData – the token structure of Block 1 for Class 5 SubClass 11 shall be as given in Table 23.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table 23 – Block 1 for Class 5 SubClass 11 meter generated token structure

Subclass	TSTN	Data		TMAC
		Data type	RES	
Binary	Binary	Binary	Binary	Binary
4 bits	10 bits	5 bits	10 bits	32 bits
11d =1011b – Meter-generated code See 6.2.3	Truncated 32-bit number	Valid values: 0d – Settlement data present in Block 2 1d – 31d **Reserved for future use**	**Reserved for future use** Reserved bits shall be zero	Truncated from 128 bits of MAC (Calculated over all participating blocks)
NOTE For Class 5 SubClass 11 TSTN is the truncated meter-generated STN.				

Similar to other sub-classes, STN is maintained by the token originator (the meter in this case) and the token originator derives TSTN to place TSTN's value in the encoding of block 1 for this SubClass.

The MAC shall not include FunctionIndex for this SubClass value.

The token structure of block 2 for Class 5 SubClass 11 is given in Table 24.

Table 24 – Block 2 for Class 5 SubClass 11 meter generated token structure

Block Sequence	Token generation month and year	Token generation day of month	Meter health	Reconciliation amount sign	Reconciliation amount config	Reconciliation amount	RES
Binary	Binary	Binary	Binary	Binary	Binary	Binary	Binary
2 bits	9 bits	5 bits	1 bit	1 bit	3 bits	18 bits	22 bit
Value: 00b See B.1.2	See 6.2.5.4.2	See 6.2.5.4.3	See 6.2.5.4.4	See 6.2.5.4.5.	See 6.2.5.4.6	See 6.2.5.4.7	Reserved for future use.

6.2.5.4.2 Token generation month and year

Token generation month and year is a 9 bit value ranging from 0d to 511d. It describes the month number since the epoch and range is sufficient for 41 years from the epoch. Epoch shall be country-specific.

For example: If the epoch is Jan 2010 then value of 1d will be Feb 2010.

6.2.5.4.3 Token generation day of month

Valid values: 0d (1st of month) to 30d (31st of month).

Values 28d to 30d representing 29th to 31st of the month are not valid if the month is of that number or fewer days.

For example: If the month is February then 30th and 31st are invalid values. If month is April, then 31st is invalid value.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.2.5.4.4 Meter health

The following are valid values:

- 0d – Healthy;
- 1d – Unhealthy: for example, a persistent tamper conditionor a detected hardware failure.

This is device-specific as to what conditions are to be taken into consideration for meter health.

The meaning of these values shall be agreed between the meter vendor and the utility.

6.2.5.4.5 Reconciliation amount sign

The following are valid values:

- 0d – Reconciliation amount is positive: utility needs to pay the amount to the consumer;
- 1d – Reconciliation amount is negative: utility needs to recover the amount from the consumer.

NOTE "Consumer" refers to the consumer occupying the premises where the meter generated the token at the time of token generation.

6.2.5.4.6 Reconciliation amount configuration

The following are valid values:

- 0d – Means Reconciliation amount is in 1/1000th of minor currency unit of country;
- 1d – Means Reconciliation amount is in 1/100th of minor currency unit of country;
- 2d – Means Reconciliation amount is in 1 minor currency unit of country;
- 3d – Means Reconciliation amount is in 10 minor currency unit of country;
- 4d – Means Reconciliation amount is in 100 minor currency unit of country;
- 5d – Means Reconciliation amount is in 1000 minor currency unit of country;
- 6d – Means Reconciliation amount is in 10000 minor currency unit of country;
- 7d – reserved for future use.

6.2.5.4.7 Reconciliation amount

The number, which when scaled according to 6.2.5.4.6, gives the amount in major currency units.

6.2.5.5 SubClass 12: RESERVED

This SubClass is reserved for future assignment by IEC TC 13/WG 15 and may not be used for any other purpose.

6.2.5.6 SubClass 13: RESERVED

This SubClass is reserved for future assignment by IEC TC 13/WG 15 and may not be used for any other purpose.

6.2.5.7 SubClass 14: RESERVED

This SubClass is reserved for future assignment by IEC TC 13/WG 15 and may not be used for any other purpose.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.2.5.8 SubClass 15: RESERVED

This SubClass is reserved for future assignment by IEC TC 13/WG 15 and may not be used for any other purpose.

6.3 Token data elements

The token data elements are given in Table 25.

Table 25 – Token data elements

Data element name	Context	Data element format	Size	Range	Reference
SubClass	Defines the type of token and whether the token is encrypted or not. These 4 bits are never encrypted.	Binary	4 bits	All possible values	6.2.3
TSTN	TokenReferenceNumber is the 10 LSB of the 32-bit STN	Binary	10 bits	All possible values	6.1.7
AMTConfig	Defines the multiplier applied to the AMT field.	Binary	2 bits	All possible values	6.2.4.1 , 6.2.5.1
AMT	Amount field defines the mantissa part of the amount and is used in combination with the AMTConfig field in representing the value fo the token.	Binary	13 bits	All possible values	6.2.4.1 , 6.2.5.1
TMAC	The least significant 32 bits of the MessageAuthenticationCode generated for the token at APDU level.	Binary	32 bits	All possible values	6.1.13, 6.1.15
CheckDigit	Check digit added to the TCDU at the final stage of generation.	Decimal	1 digit	0-9	6.2.5.3
ServiceType	Used in tokens where defined functions are initiated for SpecialFunction token.	Binary	5 bits	All possible values	6.2.5.2.2
RES	Used in token where functions are initiated for SubClass 9, SubClass 11. These bits are currently reserved for future assignment by IEC TC13 / WG 15 and shall not be used for manufacturer-specific purposes. These unused bits shall be set to 0.	Binary	10 bits	0000000000b	6.2.5.2

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.4 TCDU generation functions

The TCDU is created when the APDU is converted from a binary payload into a decimal number and offset from the other class domains and the CheckDigit is also added.

Figure 11 and Figure 12 shows how an APDU is transformed into a TCDU for unencrypted and encrypted tokens respectively.

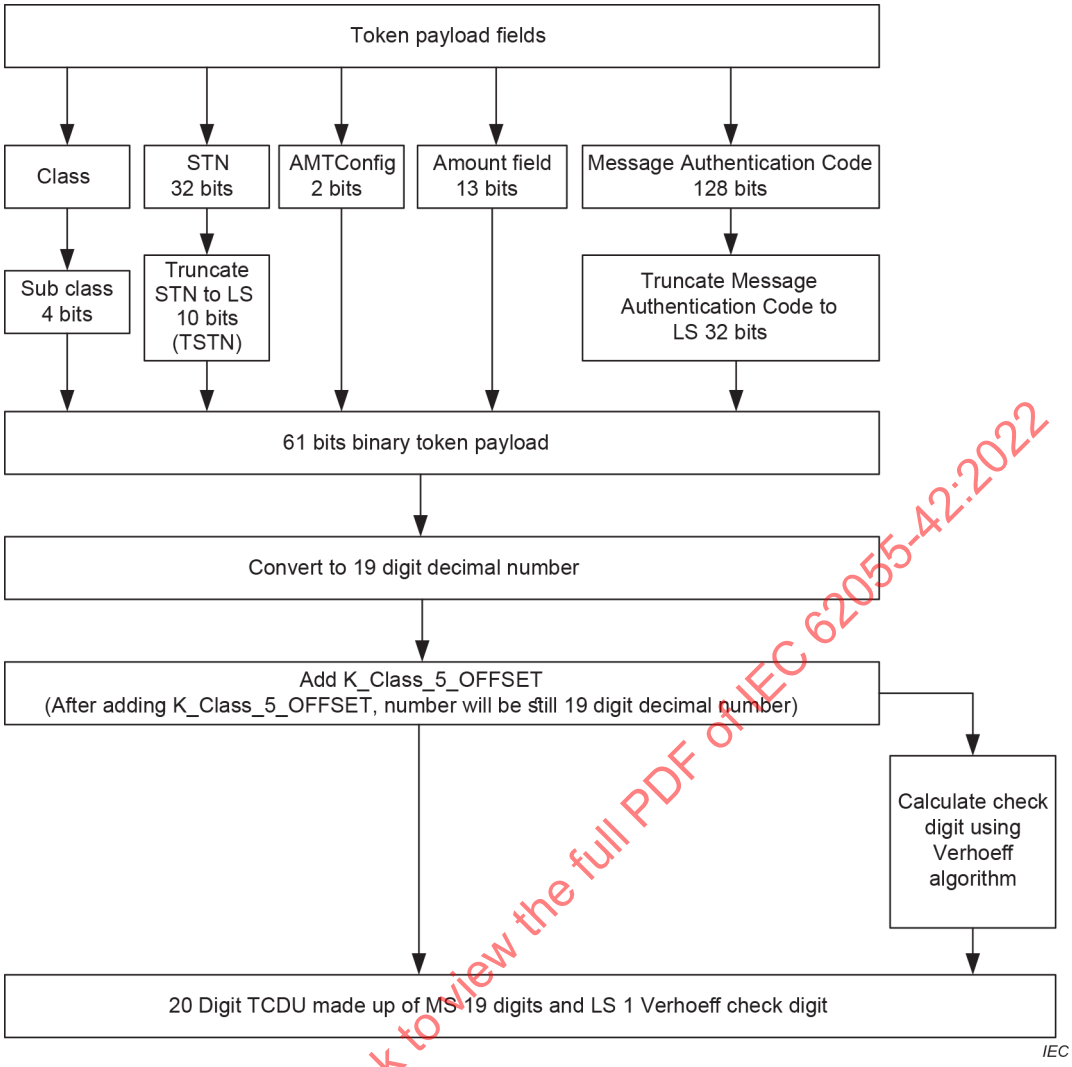
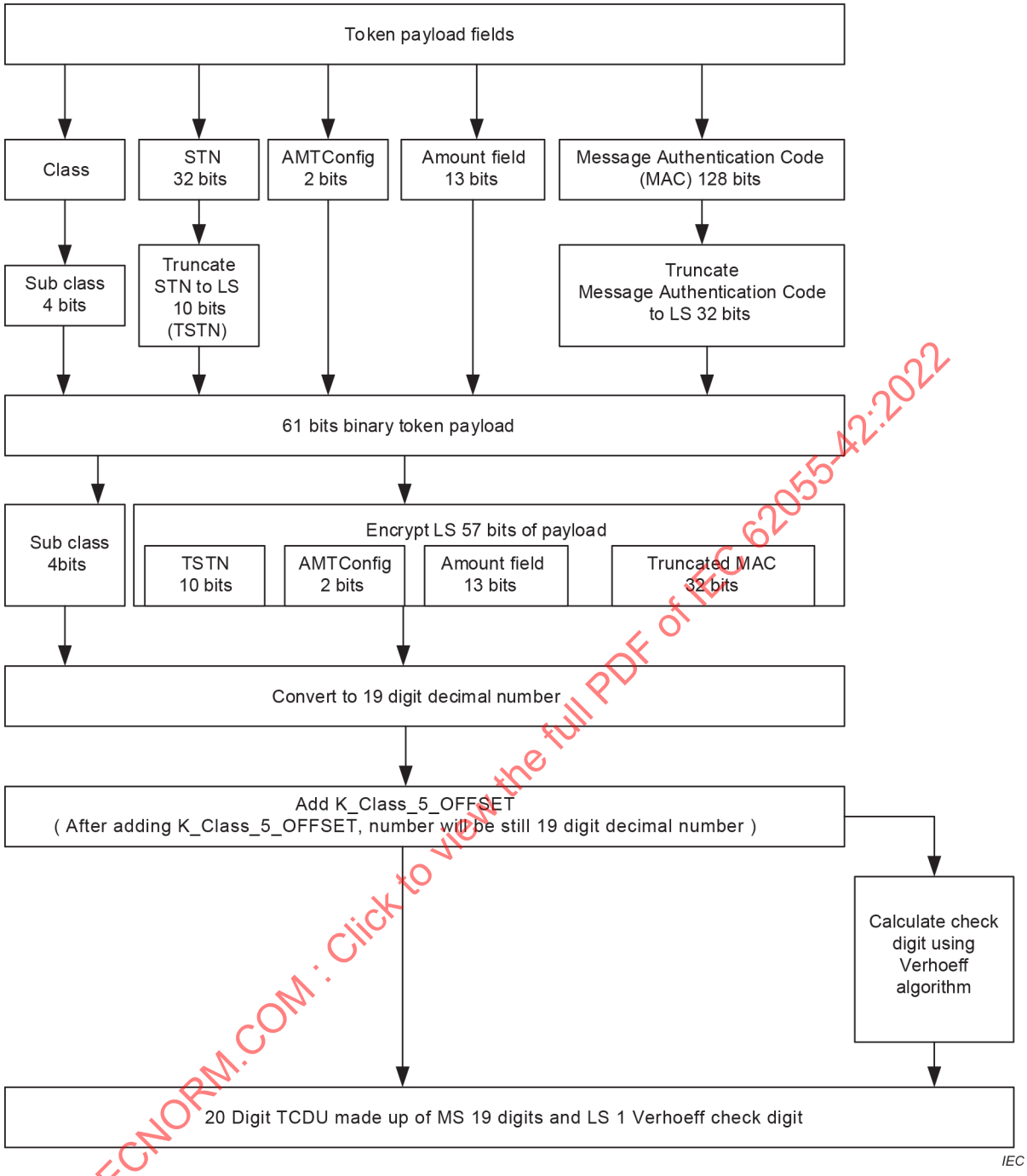


Figure 11 – TCDU generation for SubClass 0 unencrypted tokens

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022



IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Figure 12 – TCDU generation for SubClass 8 encrypted tokens

For the encryption algorithm refer to 6.5.4.

6.5 Security functions

6.5.1 General requirements

This subclause describes the encryption mechanism for tokens requiring encryption (SubClass 8-15) including the key generation and management process, key revision policy and all other matters relating to token security.

6.5.2 Key management

Key management processes are not defined in this document.

6.5.3 Key deriviation

Key derivation processes are not defined in this document.

6.5.4 Encryption process

A suitable format-preserving encryption algorithm shall be used. The format-preserving algorithm is not defined in this docment.

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 26.

Table 26 – Data elements in the APDU

Data element name	Context	Data element format	Reference
Token	The token payload is 61 bits. The 4 most signifanct bits define the type of token and whether the token is encrypted or not. These 4 bits are never encrypted.	61 bits	6.2.4.1 6.2.5.1 6.2.5.2
CheckDigit	Check digit calculated on the payload of the APDU to assist the user by detecting most keying errors.	1 digit	5.3 6.2.5.3 Annex A
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

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

7.1.2 TokenData

The TokenData from the TCDU after decryption (if relevant) and processing are presented to the MeterApplicationProcess in the APDU.

The actual 61-bit, 122-bit, 183-bit or 244-bit token (inclusive of TMAC of 32 bits) as originally entered into the APDU by the MeterApplicationProcess. The MeterApplicationProcess is now able to process it further.

7.1.3 AuthenticationResult

This is a status indicator to tell the MeterApplicationProcess that the initial authentication checks (see 7.3.3) passed or failed, in order that the MeterApplicationProcess can respond appropriately. Possible values are given in Table 27.

Table 27 – Possible values for AuthenticationResult

Data item name	Context	Format	Reference
Authentic	The authentication test passed or failed: • False if any of the tests below have failed; • True if none of the tests below have failed.	boolean	7.3.3
MACError	The checking of the MAC did not pass for any reason.	boolean	7.3.3
TokenClassError	The presented token is of a class not supported by the meter.	boolean	7.3.3
CheckDigitError	The token presented has a check digit that does not correspond to the preceding 19 digits of the token and is likely to have been entered incorrectly, or is a fraudulent token.	boolean	7.3.3

It shall optionally be left to the implementation whether to log or not to log the tokens which fail the Authentication criteria of TokenClassError and CheckDigitError to protect the meter's memory.

7.1.4 ValidationResult

This is a status indicator to tell the MeterApplicationProcess that the initial validation checks (see 7.3.4) passed or failed, in order that the MeterApplicationProcess can respond appropriately. Possible values are given in Table 28.

Table 28 – Possible values for ValidationResult

Data item name	Context	Format	Reference
Valid	The Validation test passed or failed: • False if any of the tests below have failed; • True if none of the tests below have failed.	boolean	7.3.4
OldError	The STN value as recorded in the token is older than the oldest value of recorded values in the memory of the meter, and consequently outside of the sliding window.	boolean	7.3.4
UsedError	The STN value calculated from the TSTN is in valid sliding window and then if token has already been recorded in the meter's memory.	boolean	7.3.4

7.1.5 TokenResult

After the MeterApplicationProcess has executed the instruction contained in the token, the TokenResult value reflects the outcome (see 7.3.5). 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 STN), rejection of the token, erasure of token data from the TokenCarrier, etc. Possible values are given in Table 29.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table 29 – Possible values for TokenResult

Data item name	Context	Format	Reference
Accept	The Validation test passed or failed: • False if any of the tests below have failed; • True if none of the tests below have failed.	boolean	7.3.5
OverflowError	The credit register in the meter would overflow or exceed the maximum allowable credit on the meter. When credit reduces in the meter such that the sum of "credit present in the meter" and the "credit in the token" is less than or equal to the maximum allowable credit limit, this token should be accepted.	boolean	7.3.5
RangeError/OutOfWindowError	One or more data elements in the token has a value that is outside the range of values defined in the application for that element	boolean	7.3.5
FunctionError/UnsupportedSubClassError	The particular function to execute the token is not implemented, or the SubClass of the token is not recognised by the meter.	boolean	7.3.5

7.2 APDU Extraction processes

7.2.1 APDU Extraction process for Class 5 tokens

This is a transfer process from the TCDU to the APDU and is applicable to all Class 5 tokens, despite each token SubClass being a specialisation of this process depending on the functionality. Also there is a difference between APDU Extraction of SubClass 0-7 tokens compared to SubClass 8-15, as the latter have an additional encryption step as given below.

The extraction process is as follows:

a) Verify the check digit/digits and then check digit/digits is/are removed from the TCDU (based on no. of digits in TCDU);

b) Verify the first 19 digits are in Class 5 range or not and if in range, subtract K_Class_5_Offset from TCDU from first 19 digits(as TCDU can be 20 or 40 or 60 or 80 digits inclusive of check digits);

c) Convert first 19-digit of TCDU to 61-bit binary payload and verify the SubClass of the token and determine if the token is encrypted or unencrypted from the bit number 61; if bit number 61 is 0 it means the token is unencrypted and if it is 1 it means the token is encrypted;

d) Decrypt the token if bit number 61 is having value 1b (means token is from SubClass 8-15);

e) Convert 19-digit decimal number to 61-bit binary payload (for SubClass 0, SubClass 1, SubClass 8, SubClass 9) and in the case where the TCDU is more than 19 digits (for SubClass 10), then after decryption process, decrypted binary payload is to be separated out into 61 bits of data and then to be stored to 64 bit storage blocks;

f) Decode the token according to SubClass;

g) Reconstruct STN from TSTN and check whether the token is a repeat or not, or whether STN is outside the sliding window (see 6.1.8);

h) If STN is not a repeat nor outside the sliding window, reconstruct the MAC and verify that the truncated 32 bits match with TMAC that is delivered in the token (see 6.1.15 and 6.1.16);

i) Apply the appropriate AuthenticationResult (see 7.1.3) and ValidationResult (see 7.1.4);

j) Carry out the functions and processes specified in the token APDU and apply the appropriate TokenResult (see 7.1.5).

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

7.2.2 APDU Extraction process for SubClass 0 unencrypted token

The APDU extraction process for a SubClass 0 token is shown in Figure 13.

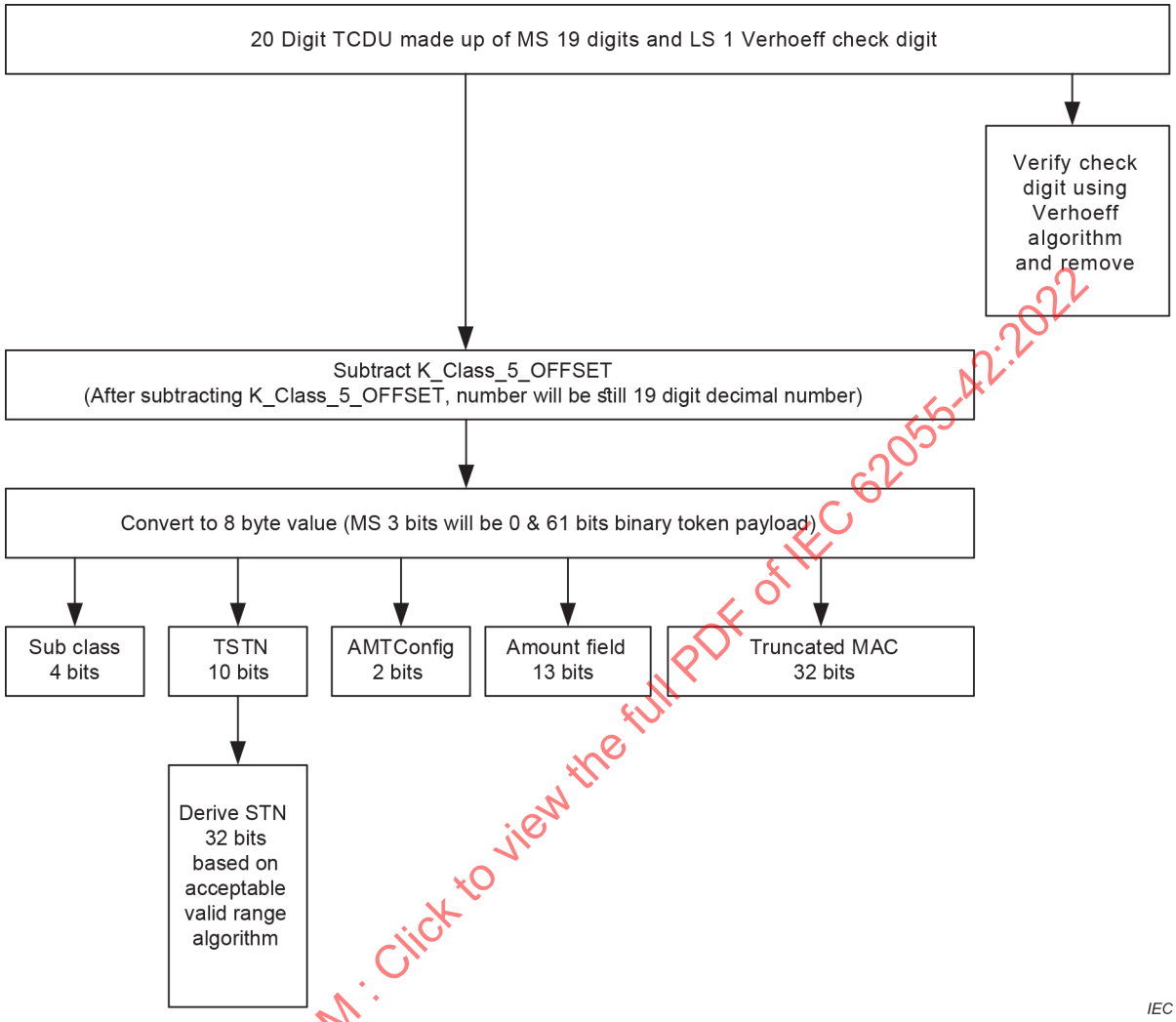
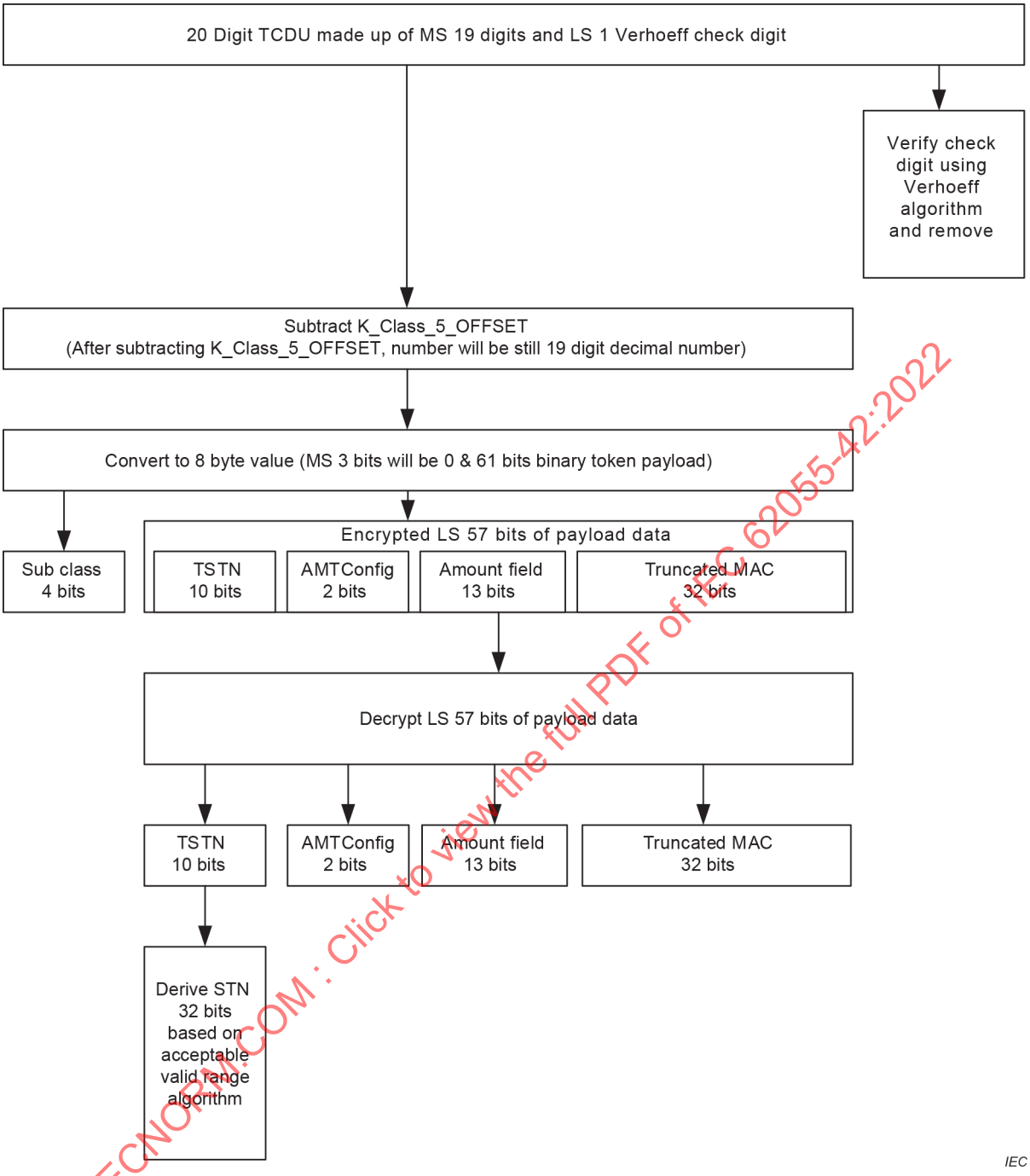


Figure 13 – APDU extraction process for SubClass 0 tokens

7.2.3 APDU Extraction process for SubClass 8 encrypted token

The APDU extraction process for a SubClass 8 token is shown in Figure 14.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022



IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Figure 14 – APDU extraction process for SubClass 8 tokens

Refer to decryption algorithm in 7.3.2.

7.3 Security functions

7.3.1 Key attributes and key changes

7.3.1.1 Key change requirements

Key change requirements are not defined in this document.

7.3.1.2 Key change processing

Key change requirements are not defined in this document.

7.3.2 Decryption algorithm

A suitable format-preserving decryption algorithm shall be used compatible with 6.5.4. The format-preserving decryption algorithm is not defined in this document.

7.3.3 TokenAuthentication

A payment meter which supports Class 5 tokens, shall on receipt of a Class 5 token, first calculate the Verhoeff check digit/digits of the token (one check digit or more check digits) based on the token SubClass by using the Verhoeff Algorithm (See Annex A for an algorithm implementation example) and match with that/those received in the token.

The following conditions shall be met:

a) If check digit/digits has/have the same value as that received in the token, AuthenticationResult status shall provisionally be considered as Authentic from the check digit point of view;

b) If above condition is not met, then AuthenticationResult status shall log CheckDigitError;

c) If the check digit test passes, then the meter application process checks whether the first 20 digits of the token belong to Class 5 range of tokens or not. If first 20 digits of the token is in the range of Class 5 tokens, AuthenticationResult status shall provisionally be considered as Authentic from Class 5 point of view;

d) If above condition is not met, then AuthenticationResult status shall log TokenClassError;

e) After the above two checks are found to be valid, the meter application layer decrypts the token if it is from one of any encrypted sub-classes. After that follows the STN deducing process, FunctionIndex derivation process and TMAC derivation process (run one after another);

f) If a valid STN is derived and a valid Function Index is derived then the MAC calculation process leads to the truncated MAC of 32-bit value. If the derived TMAC matches with that of the received token, then AuthenticationResult status shall be considered as Authentic and AuthenticationResult status shall log Authentic;

g) If above condition is not met, then AuthenticationResult status shall log MACError.

7.3.4 TokenValidation

A payment meter which supports Class 5 tokens, shall also need to process checks on the APDU elements. Main element of the APDU is STN which is partially sent in the token and then deduced at the meter-end.

The following conditions shall be met:

a) After deducing the STN value, the meter shall check whether the received token's STN is older than the oldest token record available in meter's memory i.e. outside the sliding window;

b) If the above condition is true, then TokenValidationResult status shall log as OldError;

c) If the STN value is in the sliding window range and the meter finds that it is already processed as per the record in the meter's memory, then TokenValidationResult status shall log as UsedError;

d) If none of the above indicate OldError or UsedError, then TokenValidationResult status shall log as Valid.

7.3.5 TokenResult

The token shall be rejected and TokenResult status shall not be logged as accepted if any of the following conditions are true:

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

a) If, while executing the credit amount of the token, it may lead to the payment meter credit register to overflow, TokenResult status shall log OverflowError and token shall be rejected and shall not be further processed;

b) In the case where the encoding data of the token APDU do not comply with the definitions given as per the SubClass or the values are outside of the defined range of values defined in the APDU encoding of a particular SubClass, then TokenResult status shall log RangeError/OutOfWindowError;

c) If, while executing the instruction of the token, any of the parameters are not implemented by the meter, then TokenResult status shall log FunctionError/UnsupportedSubClassError;

d) If none of the above errors are true, then TokenResult status shall log Accept.

It shall optionally be left to the implementation which of the following methods to use:

e) Only log/indicate whether token is accepted or rejected;

f) Log/indicate token acceptance or rejection with detailed reason of error.

8 MeterApplicationProcess requirements

8.1 General requirements

In addition to the requirements given in 7.3.5 the MeterApplicationProcess shall execute tokens in accordance with the definitions given in Clause 6 and Clause 7 and shall manage all communicatons interfaces below the TCDU as needed and are assumed to be outside the scope of this document.

8.2 Token acceptance/rejection

All compliant meters shall be capable of reading, interpreting and executing some or all the token types successfully. Sucessful reading and interpreting of such tokens could result in a rejection if the token type is not supported.

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

- AuthenticationResult indicates a status value of Authentic in the APDU (see 7.1.3);
- ValidationResult indicates a status 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 exception and proviso:

- Successful processing of the first entered token (and subsequent token or tokens if not the final token) of a token group shall not indicate Accept, but it shall indicate provisional acceptance until the final token of the token group is also accepted;
- Successful processing of the final entered token of a token group 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 CheckDigitError in the APDU (see 7.1.3);
- AuthenticationResult indicates a status value of ClassError 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);

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

- 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 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 application of 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 MeterApplicationProcess cannot execute the token (see 8.2);
- if accepting the credit on a token would cause the credit register to overflow (see 8.2).

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 status of the presented TCDU from the user interface immediately within few seconds of TCDU conveyance. The minimum support of status of the TCDU presenting to the meter will be ACCEPTED or REJECTED message.

TCDU processing result status on demand is OPTIONAL.

If the TCDU gets rejected by the payment meter then it is OPTIONAL to give feedback on the display with a more detailed rejection reason.

Indicators relating to the result of token entry shall only be displayed on the 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 6.2.4.1 and 6.2.5.1 for more detail on the structure of these tokens.

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 as indicated by the SubClass field in the token.

8.5 Engineering/SpecialFunction tokens

See 6.2.4.2 and 6.2.5.2 for more detail on the structure of these tokens.

All payment meters may support any or all of these tokens. If any of the incorporated functions are not supported the payment meter shall perform the sub-set of functions that are supported.

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

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

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

9 KMS: KeyManagementSystem generic requirements

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

Due to the variability of key management requirements by geography and project, and to some degree reliance on different end-to-end security protocols, it is recommended that such processes are described in detail in companion specifications tailored to the market requirements.

10 Maintenance of unassigned entities

Assignment of currently unassigned entities in this document such as reserved sub-classes shall be exclusively managed by IEC TC 13 Working Group 15, through the normally instantiated New Proposal processes of IEC.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Annex A
(informative)

Verhoeff code implementation example

A.1 Sample code

Annex A is an informative annex giving a possible code implementation of the Verhoeff algorithm written in Delphi. This code is included on a RANDz basis and is intended to be used as a guideline for implementers and not as a normative part of this document.

Below is some straightforward code (in Delphi) for demonstrating the Verhoeff algorithm.

```
{ UTRNrec is an array[1..20] of byte.
    It is important to note that 1 is the leftmost digit as normally read
     and written – i.e. expected to be in the 7 to 9 range.
     20 is where the check digit goes. Do NOT change the order!
     If you have to change to indexing 0 to 19 then so be it. Just because
      C never got the idea that humans might sometimes like arrays to start
      with an index other than 0 and it was the compiler's job to sort it out.
    The elements are of course the values of the digits not ASCII characters.
}

{ These routines do not check that the values in U are all in the range 0..9,
   but this shall be the case, so a prior check of this condition is assumed.
}

const
{ 8 rows, 10 columns.
  p is Verhoeff's standard permutation table. See Verhoeff (1969) (p. 95).
  Of order 8, so p[i, j] = p[1, p[i-1, j]] for i in [2..7], j in [0..9]
   and it could be extended indefinitely by repeating these 8 rows.
  This does not detect some phonetic errors 15-50 and 17-70.
  Effectively it detects 100% of phonetic errors in 15 places
    and over 80% of them in the other 4 places out of the 19 possible.
}
  p: array[0..7, 0..9] of integer =
    (
      ( 0,1,2,3,4,5,6,7,8,9 ),
      ( 1,5,7,6,2,8,3,0,9,4 ),
      ( 5,8,0,3,7,9,6,1,4,2 ),
      ( 8,9,1,6,0,4,3,5,2,7 ),
      ( 9,4,5,3,1,2,6,8,7,0 ),
      ( 4,2,8,6,5,7,3,9,0,1 ),
      ( 2,7,9,3,8,0,6,4,1,5 ),
      ( 7,0,4,6,9,1,3,2,5,8 )
    );  { Table 14.8.a }

{ The multiplication table for the dihedral group of order 10 }
  d: array[0..9, 0..9] of integer =
    ( ( 0,1,2,3,4,5,6,7,8,9 ),
      ( 1,2,3,4,0,6,7,8,9,5 ),
      ( 2,3,4,0,1,7,8,9,5,6 ),
      ( 3,4,0,1,2,8,9,5,6,7 ),
      ( 4,0,1,2,3,9,5,6,7,8 ),
      ( 5,9,8,7,6,0,4,3,2,1 ),
      ( 6,5,9,8,7,1,0,4,3,2 ),
      ( 7,6,5,9,8,2,1,0,4,3 ),
      ( 8,7,6,5,9,3,2,1,0,4 ),
      ( 9,8,7,6,5,4,3,2,1,0 ) );  { Table 14.8.b }

  final_array:   array [0..9] of integer =
      ( 1, 2, 6, 7, 5, 8, 3, 0, 9, 4 );  { Table 14.8.c }

function UTRNCheckDigitGBCSFromFirst19(const U: UTRNrec): integer;
```

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

```
var
  J, K, L, IntDig, CurDig: integer;
begin
  IntDig:= 0;
  K:= 4;  { choose carefully which permutation to apply to the first digit }
  for J:= 1 to 19 do
    begin
      CurDig:= U[J];
      L:= p[K, CurDig];
      if K < 7 then K:= K+1 else K:= 0;
      IntDig:= d[IntDig, L];
    end;
  IntDig:= final_array[IntDig];
  UTRNCheckDigitGBCSFromFirst19:= IntDig;
end;

procedure SetUTRNCheckDigitGBCS(var U: UTRNrec);
{ Inserts the correct check digit at U[20] }
begin
  U[20]:= UTRNCheckDigitGBCSFromFirst19(U);
end;

function UTRNChecksGBCS1(const U: UTRNrec): boolean;
{ If the check digit is valid then this returns TRUE.
  If the check digit is not valid then this returns FALSE.
  Method – comparison of check digit with computed checksum for first 19 digits.
}
begin
  UTRNChecksGBCS1:= ( U[20] = UTRNCheckDigitGBCSFromFirst19(U) );
end;

function UTRNChecksGBCS2(const U: UTRNrec): boolean;
{ If the check digit is valid then this returns TRUE.
  If the check digit is not valid then this returns FALSE.
  Method – computation of check of all 20 digits
}
var
  J, K, L, IntDig, CurDig: integer;
begin
  IntDig:= 0;
  K:= 4;  { choose carefully which permutation to apply to the first digit }
  for J:= 1 to 20 do
    begin
      CurDig:= U[J];
      L:= p[K, CurDig];
      if K < 7 then K:= K+1 else K:= 0;
      IntDig:= d[IntDig, L];
    end;
  UTRNChecksGBCS2:= (IntDig=0);
end;
```

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

For a SuperToken, UTRNCheckDigitGBCS function should be modified such that it returns CheckDigit for passed array pointer and number of digits for which CheckDigit is to be derived.

Annex B
(informative)

Example of ExtendedTransferCredit

B.1 Class 5: SubClass 10: TransferCredit + Tariff

B.1.1 General

In the diagrams and tables describing the token and block structures, all fields are considered as binary sub-fields of the block-forming part of the token. For clarity, the check digit which is appended to the decimal representation of the block is not shown. The number of bits is shown for each field. In most cases the field is interpreted as its binary value. Fields representing bit masks or choices are described accordingly.

The generic structure for the Class 5 SubClass 10 token is given in Table B.1, Table B.2, Table B.3 and Table B.4:

Table B.1 – Block 1 of TransferCredit + tariff token

SubClass	TSTN	Data		TMAC
		AMTConfig	AMT	
4 bits	10 bits	2 bits	13 bits	32 bits
10d = 1010b See 6.2.3	See NOTE 2			See NOTE 2 and NOTE 3

NOTE 1 For class 5 SubClass 10 tokens, the MS 4 bits of the first block (the SubClass) are transmitted unencrypted and the MS bit is used to determine whether all blocks of the token are encrypted or unencrypted.

NOTE 2 The TSTN, AMTConfig, AMTand MAC field valid values and their interpretations are the same as those of the corresponding fields of the token of Class 5 SubClass 8 (see 6.2.5.1).

NOTE 3 The MAC field is truncated from 128 bits of GMAC calculated over all participating blocks. The method of calculating the 128-bit GMAC is detailed in 6.1.13.

Table B.2 – Block 2 of TransferCredit + Tariff token

Block Sequence	Data				
	Complete tariff	Tariff type	Tariff sub info	Tariff activation month	Tariff data
Binary	Binary	Binary	Binary	Binary	Binary
2 bits	1 bit	4 bits	4 bits	6 bits	44 bits
Valid values: 0d to 2d See B.1.2	Valid values: 0d, 1d See B.1.3	Valid values: 0d to 5d See Table B.5	See B.1.4	See B.1.5	Detailed separately for each type of SubClass 10 token

B.1.2 Block sequence/SuperTokenBlockToFollow

Block sequence appears in the second and subsequent blocks of a multi-block token. It is the MS 2 bits of the block and is transmitted unencrypted. Valid values are given as follows:

- Block sequence 0d – this is the last 20-digit block;
- Block sequence 1d – one 20-digit block is to follow this block;

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

- Block sequence 2d – two 20-digit blocks are to follow this block;
- Block sequence 3d – NOT USED.

NOTE This allows a device to determine, without decryption of an encrypted token, whether more blocks are expected by inspection of the decimal value of the second and following blocks.

B.1.3 Complete tariff

One or more TransferCredit+Tariff tokens may be required to transmit all the information required in a complete tariff or tariff update. In a TransferCredit+Tariff token, which is the only token required, or is the last of a sequence of tokens required, this bit is 1. For tokens other than the last token in a sequence, this bit value remains 0b.

B.1.4 Tariff sub-information

The interpretation of tariff sub-information depends on the value of the tariff type as is described below.

B.1.5 Tariff activation month

The tariff activation month instructs the meter that the tariff should be activated either immediately on completed receipt of the tariff or as soon as possible after a specified month has begun.

NOTE The encoding permits specification of a month which is unambiguous within a period of 5 years.

The tariff activation date may be other than the start of the month if that is appropriate for a country-specific reason.

Valid values for tariff activation month are: 0d to 60d; 61 to 63 are invalid values.

Activation month value 0d indicates immediate tariff activation.

Rest of the values for month number 1d to 60d is month number which is unambiguous within a period of 5 years.

One of the ways on how to use tariff activation month is as follows.

To enable the meter to detect whether the tariff needs to be activated or not, the below data is derived by the meter:

- At the time of commissioning, the token will have activation month always immediate; i.e value is 0. On receipt of a valid token of commissioning with activation month 0, the meter will derive the month number as month number since Jan 2010 based on meter's time. For example: the epoc always rolls over every 5 years; i.e. 2010, 2015, 2020, etc., so 60 month offset applies to the latest resolved epoc;
- Then the tariff activation month is conveyed to the meter either immediate or month number of index of 5 years starting from 1d to 60d. For example: if Meter is commissioned in month of Oct 2018, the meter will store reference month value (12 * ({2017-2010} + 1) +10) = 106 (decimal), which unambiguous value of the 5-year cycle is 106 MOD 60 = 46 (decimal);
- The tariff activation month conveyed to the meter shall be in the range 46 to 60 or 0 to 34 as it is considered here that the tariff gets revised at least once in 4 years and the duration between two tariff revisions will not be more than 48 months. So, in such a case the meter will get a new tariff at least once in 4 years' duration. There can be other methods on how to use the tariff activation month using unambiguous month of 5-year's duration for tariff activation purposes.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

B.1.6 Tariff data

The structure and interpretation of the tariff data field depends on the values of the tariff type and tariff sub-information fields as is described in Table B.3 and Table B.4 for each type of tariff token.

Table B.3 – Block 3 of TransferCredit + Tariff token

Block sequence	Data
2 bits	59 bits
Value: 0d or 1d	Data interpreted according to tariff type and tariff sub-information
NOTE The block sequence value is 1 if and only if a fourth block is present in the token.	

Table B.4 – Block 4 of TransferCredit + Tariff token

Block sequence	Data
2 bits	59 bits
Value: 0d	Data interpreted according to tariff type and tariff sub-information

B.1.7 Tariff types

Each Class 5 SubClass 10 (TransferCredit + Tariff) token represents a type of tariff as given in Table B.5.

Table B.5 – Tariff types

Tariff type value	Tariff type description	Remark
0d	Slab tariff or time of use (TOU) tariff	Definitions of either slab sizes or time periods are sent
1d	Rate prices and fixed charge prices	Definitions of either energy prices or fixed charge prices are sent
2d	Electricity duty	
3d, 4d, 5d	Reserved for companion standard	
6d to 15d	Reserved	

B.1.8 Tariff sub-information

Each tariff type can be used to send several related sets of parameters to a meter. The 4-bit field tariff sub-information is interpreted as a bit mask specifying which of the sets of parameters are present in the tariff. When only one set can be sent in a token, not more than one bit of tariff sub-information is set.

For example: Either slab tariff parameters or time of use tariff parameters can be sent in a token. Therefore, if tariff type is 0, the only valid values of tariff sub-information are 0001b and 0010b.

Details of tariff sub-information are given in Table B.6.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table B.6 – Details of tariff sub-information

Tariff type	Tariff type description	Tariff sub-information	Parameter description	Remarks on tariff sub-information value
0	Slab tariff or time of use tariff	 Bit 0 Bit 1 Bits 2, 3	 Slab tariff Time of use tariff Reserved	Only one bit can be set in any token
1	Rate and fixed charge prices	 Bit 0 Bit 1 Bits 2, 3	 Rate prices Fixed charge prices Reserved	Only one bit can be set in any token Energy charges
2	Electricity duty	 Bit 0 Bits 1 to 3	 Electricity duty Reserved	Only one bit can be set in any token
3 to 15	Reserved			

Parameters have been grouped into the different sub-types in view of the following considerations:

- Energy and fixed charge prices change most frequently;
- Slab tariff details and time-of-use tariff details change less frequently than energy prices and fixed charge prices;
- Electricity duty changes infrequently and is separated from other parameters;
- Slab tariff or time-of-use tariff, energy and fixed charge prices, and duty parameters are the same for all consumers in a single supplier category.

B.2 Class 5, SubClass 10, tariff type 0: TransferCredit + slab or time-of-use tariff

B.2.1 Class 5, SubClass 10, tariff type 0, sub-type 0: TransferCredit + slab tariff

Block 1: TransferCredit – Refer Table B.1.

Block 2: Slab tariff – The token structure of Block 2 for class 5, SubClass 10, tariff type 0, sub-type 0 is given in Table B.7 and Table B.8.

Table B.7 – Block 2 for class 5, SubClass 10, tariff type 0, sub-type 0 (TransferCredit + slab tariff)

Block sequence	Complete tariff	Tariff type	Tariff sub-info	Tariff activation month	Data
2 bits	1 bit	4 bits	4 bits	6 bits	44 bits
Value: 0 or 1 See B.1.2	See B.1.3	0000b See B.1.7	0001b See B.1.8	See B.1.5	See Table B.8 Table B.8

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table B.8 – Block 2 for Class 5, SubClass 10, tariff type 0, sub-type 0 (TransferCredit + slab tariff) – tariff data part

Data					
Number of slab boundaries	**Slab scaling**	**Slab field size**	**Slab value 1**	**Slab value 2**	**Slab value 3**
4 bits	2 bits	2 bits	12 bits	12 bits	12 bits
Valid values: 0 to 7 See B.2.2	Valid values: 0 to 3 See B.2.3	Valid values: 0 to 3 (Table B.8 slab values are shown based on value 3 here) See B.2.4	0 to 4095 See B.2.5	0 to 4095 See B.2.5	0 to 4095 See B.2.5
NOTE The structure shown is for the 12-bit slab value fields (slab field size = 3). In this case, if not more than 3 slab values are sent (Number of slab boundaries <= 3) then this will be the last block, otherwise the third block will be required.					

B.2.2 Number of slab boundaries

This field informs the meter how many slabs are in the tariff (one more than the number in this field).

Valid values are as follows:

- 0 – flat rate, i.e. no slab boundaries; in effect, one slab of infinite size;
- 1 to 7 – 2 to 8 slabs present in tariff and 1 to 7 values conveyed in token;
- 8 to 15 – reserved.

Fields in the token specified for slabs not present in the tariff are ignored and need not be sent. For example: if the "number of slab boundaries" field is 2, slab values 1 to 3 are sent and slab values numbered 4 and higher may be ignored; block 3 is unnecessary as slab values 1 to 3 fit into block 2.

B.2.3 Slab scaling

This field informs the meter how the slab value sent is to be converted to kWh (or to kVAh in the case of a kVAh tariff).

Valid values are as follows:

- 0 – kWh = value × 1;
- 1 – kWh = value ×10;
- 2 – kWh = value ×100;
- 3 – reserved.

B.2.4 Slab field size

This field informs the meter how many bits are used to transmit each slab value sent.

Valid values are as follows:

- 0 – slab value in 6 bits, 6 values can be transmitted in block 2;
- 1 – slab value in 7 bits, 5 values can be transmitted in block 2;
- 2 – slab value in 9 bits, 4 values can be transmitted in block 2;

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

- 3 – slab value in 12 bits, 3 values can be transmitted in block 2.

The slab values are packed according to the slab field size, without crossing a block boundary. The structure shown is for the 12-bit field size, so 3 values can be sent in block 2 with possible values 0 to 4095 (in practice 0 is not used), and they occupy 36 bits. If more than 3 slab values are sent then block 3 will be transmitted. If the slab value is in 6 or 7 bits then 5 values can be sent in block 2 (in the space shown for slab values 1, 2, and 3) and block 3 is never required. If the slab value is in 9 bits then block 3 is only required in the case of 5 values being sent as 4 values can be packed into block 2.

B.2.5 Slab value

This field in conjunction with slab scaling (see B.2.3), determines the boundary of each tariff slab in energy terms. The starting point for each slab is the sum of the sizes of all lower slabs.

Block 3: Slab tariff – The token structure of block 3 for Class 5, SubClass 10, tariff type 0, sub-type 0 is given in Table B.9.

Table B.9 – Block 3 for class 5, SubClass 10, tariff type 0, sub-type 0 (TransferCredit + slab tariff)

Block sequence	Slab value 4	Slab value 5	Slab value 6	Slab value 7	RES
2 bits	12 bits	12 bits	12 bits	12 bits	11 bits
Value: 0 See B.1.2	0 to 4095 See B.2.5	0 to 4095 See B.2.5	0 to 4095 See B.2.5	0 to 4095 See B.2.5	Reserved
NOTE Table B.9 Slab values are shown based on value 3 of field "Slab field Size" of Table B.8.					

B.2.6 Class 5, SubClass 10, tariff type 0, sub-type 1: TransferCredit + time of use (TOU) tariff

The structure shown is an efficient representation of time of use tariffs having up to 8 rates, 2 day types, and half-hour (30 min) resolution. More rates or day types, or finer resolution, would require a different encoding.

Block 1: TransferCredit – Refer Table B.1.

Block 2: Time of use tariff – The token structure of Block 2 for Class 5, SubClass 10, tariff type 0, sub-type 1 is given in Table B.10.

Table B.10 – Block 2 for class 5, SubClass 10, tariff type 0, sub-type 1 (TransferCredit + time of use tariff)

Block sequence	Complete tariff	Tariff type	Tariff sub-info	Tariff activation month	Weekday definitions	Time periods	Res
2 bits	1 bit	4 bits	4 bits	6 bits	7 bits	35 bits	2 bits
Value: 2 or 1 See B.1.2	See B.1.3	0000b See B.1.7	0010b See B.1.8	See B.1.5	Bit string. See B.2.7	Bit string. See B.2.8	Reserved

B.2.7 Week definition

This is a 7-bit field according to which the meter allocates a day type to each day of the week.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

The following conditions shall be met:

- Bit 0 to bit 6 represent Monday to Sunday respectively;
- When the bit is 0 the corresponding day has day type 1;
- When the bit is 1 the corresponding day has day type 2.

The number of day types present in the tariff can be deduced from the value of this field as follows:

- If it is 0000000b (0) then all days have day type 1 and only one day type (day type 1) is present;
- If it is any value from 0000001b to 1111110b (1 to 126) then two day types are present;
- The value 1111111b (127) is reserved and is not used.

B.2.8 Time period definitions

The maximum number of time-of-day periods supported is 8, with half-hour resolution. The 35-bit field used for this can define these, encoded as follows: proceed along the bit string from MSB to LSB (left to right). Take 2 bits to define both the number of data bits following them and a value to be added to the normal interpretation of their value as binary, to obtain a duration in half-hours, as follows:

- 00b – 2 bits follow; add 1 to their value, hence process 4 bits in total, representing values 1 – 4 half-hours as 0000b – 0011b;
- 01b – 2 bits follow; add 5 to their value, hence process 4 bits in total, representing values 5 – 8 half-hours as 0100b – 0111b;
- 10b – 3 bits follow; add 9 to their value, hence process 5 bits in total, representing values 9 – 16 half-hours as 10000b – 10111b;
- 11b – 5 bits follow; add 17 to their value, hence process 7 bits in total, representing values 17 – 48 half-hours as 1100000b – 1111111b.

The first duration is that of period 1, which is taken to start at midnight. This gives the start time of period 2. The duration of period 2 then gives the start time of period 3, and so on. The eighth duration need not be transmitted as if the first seven durations add up to less than 48 half-hours then the eighth duration makes up the remainder of the day and there are eight periods. The number of periods need not be transmitted as if seven or fewer durations add up to 48 half-hours the day has been completed and the number of periods is thereby given.

It is to be understood that the processing of the bit field should stop when either the whole day has been covered, eight periods have been defined, or 35 bits have been processed. Fewer than 35 bits are required for many combinations of eight or fewer durations, but 35 bits suffice in all cases.

Block 3: Time of use tariff – The token structure of block 3 for class5, SubClass 10, tariff type 0, sub-type 1 is given in Table B.11.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table B.11 – Block 3 for class 5, SubClass 10, tariff type 0, sub-type 1 (TransferCredit + time of use tariff)

Block Sequence	D1R1	D1R2	D1R3, D1R4, D1R5, D1R6	D1R7	D2R1	D2R2	D2R3, D2R4, D2R5, D2R6	D2R7	ToU RES 1
2 bits	4 bits	4 bits	4 bits each (16 bits)	4 bits	4 bits	4 bits	4 bits each (16 bits)	4 bits	3 bits
Value: 0 or 1	See B.2.9.								Reserved
NOTE 1 To save space in laying out this table, the columns for D1R3, D1R4, D1R5, D1R6, and for D2R3, D2R4, D2R5, D2R6 are not shown separately, but these items are to be understood to occupy two groups of four consecutive 4-bit fields where shown. NOTE 2 If there are 7 or fewer time of use periods, this will be the final block in the tariff and the value of block sequence will be 0. If there are 8 time of use periods, the fourth block will be used to transmit the register definitions for period 8 and the value of block sequence will be 1.									

B.2.9 Register definitions

The fields labelled D1R1 … D2R8 have values from 0 to 15 and select a tariff rate register numbered 1 to 16 according to the day type (1 or 2) and time of day (period 1 to 8). For example: if D1R6 has the value 3, then the tariff rate register numbered 4 will be active if the day type is 1 and the time of use period is 6.

Block 4: Time of use tariff – the token structure of block 4 for Class5, SubClass 10, tariff type 0, sub-type 1 is given in Table B.12.

Table B.12 – Block 4 for class 5, SubClass 10, tariff type 0, sub-type 1 (TransferCredit + time of use tariff)

Block Sequence	D1R8	D2R8	RES
2 bits	4 bits	4 bits	51 bits
Value: 0 See B.1.2	See B.2.9	See B.2.9	Reserved
NOTE This block will not be present if fewer than 8 time-of-use periods are required.			

B.3 Class 5, SubClass 10, tariff type 1: TransferCredit + rate prices or fixed charge price token format

B.3.1 Class 5, SubClass 10, tariff type 1: tariff sub-information

For tariff type 2, tariff sub-information informs the meter whether the tariff information contains energy rate prices or fixed charge prices. Only one bit may be set as follows:

- 0000b – invalid;
- Bit 0 set: value is 0001b – energy rate prices;
- Bit 1 set: value is 0010b – fixed charge prices;
- 0011b to 1111b – invalid.

B.3.2 Class 5, SubClass 10, tariff type 1, sub-type 0: TransferCredit + rate prices

Block 1: TransferCredit – Refer Table B.1Block 2: Rate prices – the token structure of Block 2 for Class 5, SubClass 10, tariff type 1, sub-type 0 is given in Table B.13 and Table B.14.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table B.13 – Block 2 for class 5, SubClass 10, tariff type 1, sub-type 0 (TransferCredit + rate prices)

Block sequence	Complete tariff	Tariff type	Tariff sub info	Tariff activation month	Data
2 bits	1 bit	4 bits	4 bits	6 bits	44 bits
Value: 0 or 1 or 2 See B.1.2	See B.1.3	0001b	0001b See B.3.3 See B.1.8	See B.1.5	See Table B.14

Table B.14 – Block 2 for class 5, SubClass 10, tariff type 1, sub-type 0 (TransferCredit + rate prices) – tariff data

Data					
Number of rate prices	**Rate price multiplier**	**Rate price field size**	**Rate price 1**	**Rate price 2**	**RES**
4 bits	3 bits	2 bits	14 bits	14 bits	7 bits
Valid values: 0 to 7 See B.3.4	Valid values: 0 to 6 See B.3.5	Valid values: 0 to 3 See B.3.6 (Table B.14, Table B.15, Table B.16 rate price value is shown based on value 3 of this field)	0 to 16383 See B.3.7	0 to 16383 See B.3.7	Reserved
NOTE 1 The structure shown corresponds to rate price field size having the value 3, signifying 14-bit rate price fields. NOTE 2 If there are only 1 or 2 rate prices sent (number of rate prices field has value 0 or 1), they will both fit in this second block, so the next (third) block is not needed and block sequence will be 0, If there are more than 2 rate prices to be sent the third block will be needed and block sequence will be 1.					

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

B.3.3 Class 5, SubClass 10, tariff type 1: tariff sub-information

Tariff sub-information is a 4-bit field describing what is present in the token. Valid values have one and only one bit set as follows:

- Bit 0 – Energy rate prices (value 0001b = 1);
- Bit 1 – Fixed charge prices (value 0010b = 2);
- Bits 2 and 3 – reserved.

B.3.4 Number of rate prices

This field defines the number of rate prices sent in the tariff. Rate price fields for unused rates shall be set to zero or absent (for example, if 2 rate prices are sent, the last block may be absent).

Valid values are as follows:

- 0 – flat rate, i.e. single rate price present in tariff;
- 1 to 7 – 2 to 8 rate prices present in tariff;
- 8 to 15 – reserved.

B.3.5 Rate price multiplier

This is a scaling factor by which values in the rate price fields are multiplied to obtain the required monetary value in minor currency units. Valid values are as follows:

- 0 = × 0,001;
- 1 = × 0,01;
- 2 = × 0,1;
- 3 = × 1;
- 4 = × 10;
- 5 = × 100;
- 6 = × 1 000;
- 7 = reserved.

B.3.6 Rate price field size

This field allows the rate price information to be sent in fields having a different maximum value as follows:

- 0 – 11 bits (0..2047);
- 1 – 12 bits (0..4095);
- 2 – 13 bits (0..8191);
- 3 – 14 bits (0..16383).

B.3.7 Rate price value

This field in conjunction with Rate price multiplier (see B.3.5), determines the rate price value in minor currency.

Block 3: Rate prices – the token structure of Block 3 for Class5, SubClass 10, tariff type 1 is given in Table B.15.

Table B.15 – Block 3 for class 5, SubClass 10, tariff type 1, sub-type 0 (TransferCredit + rate prices)

Block Sequence	Data				
	Rate Price 3	Rate Price 4	Rate Price 5	Rate Price 6	RES
2 bits	14 bits	14 bits	14 bits	14 bits	3 bits
Value: 0 or 1 See B.1.2	0 to 16383 See B.3.7	0 to 16383 See B.3.7	0 to 16383 See B.3.7	0 to 16383 See B.3.7	Reserved
NOTE The structure shown corresponds to rate price field size having the value 3, signifying 14-bit rate price fields.					

Block 4: Rate prices – The token structure of Block 4 for Class5, SubClass 10, tariff type 1 is given in Table B.16.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table B.16 – Block 4 for class 5, SubClass 10, tariff type 1, sub-type 0 (TransferCredit + rate prices)

Block Sequence	Data		
	Rate Price 7	Rate Price 8	RES
2 bits	14 bits	14 bits	31 bits
Value: 0 See B.1.2	0 to 16383 See B.3.7	0 to 16383 See B.3.7	Reserved
NOTE The structure shown corresponds to rate price field size having the value 3, signifying 14-bit rate price fields.			

B.3.8 Class 5, SubClass 10, tariff type 1, sub type 1: TransferCredit + fixed charge prices

Block 1: TransferCredit – Refer Table B.1.

Block 2: Fixed charge prices – the token structure of block 2 for Class 5, SubClass 10, tariff type 1 is given in Table B.17 and Table B.18.

Table B.17 – Block 2 for class 5, SubClass 10, tariff type 1, sub-type 1 (TransferCredit + fixed charge prices)

Block sequence	Complete tariff	Tariff type	Tariff sub info	Tariff activation month	Data
2 bits	1 bit	4 bits	4 bits	6 bits	44 bits
Value: 0d See B.1.2	See B.1.3	0001b See B.1.7	0010b See B.3.3 See B.1.8	See B.1.5	See Table B.18
NOTE The structure shown is applicable to 12-bit fixed charge fields, i.e. fixed charge price field size has the value 2.					

Table B.18 – Block 2 for class 5, SubClass 10, tariff type 1, sub-type 1 (TransferCredit + fixed charge prices) – tariff data

Tariff data					
Number of fixed charge prices	Fixed charge price multiplier	Fixed charge price field size	Fixed charge application	Fixed charge price 1 value	RES
4 bits	3 bits	2 bits	3 bits	12 bits	20 bits
Valid values: 0d See B.3.9	Valid values: 0d to 5d See B.3.10	Valid values: 0d to 3d See B.3.11 (Table B.18 Fixed charge proce1 value is shown based on value 2 of this field)	Valid values: 0d to 2d See B.3.12	See B.3.13	Reserved

B.3.9 Number of fixed charge prices

This field defines the number of fixed charge prices in the tariff as follows:

- 0d – Means 1 fixed charge is present in the tariff;

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

- 1d to 15d – reserved.

B.3.10 Fixed charge price multiplier

This field defines a multiplier to be used in conjunction with the value of a fixed charge price field to obtain a price in minor currency units. The interpretation of the value is the same as for the rate price multiplier (see B.3.5).

B.3.11 Fixed charge price field size

This field defines the size of each fixed charge price field as follows:

- 0 – 8 bits;
- 1 – 10 bits;
- 2 – 12 bits;
- 3 – reserved.

B.3.12 Fixed charge application

This field defines how the fixed charge is applied as follows:

- 0 – per billing period;
- 1 – per kW or kVA of consumer sanctioned load / bill duration;
- 2 – daily charge;
- 3 to 7 – reserved.

B.3.13 Fixed charge price value

The value in this field, when multiplied by the scaling factor (see B.3.10), gives the monetary value of the fixed charge price in minor currency.

B.4 Class 5, SubClass 10, tariff type 2: TransferCredit + electricity duty (ED) token format

B.4.1 Electricity duty (ED)

In some countries a duty or tax is levied (the words tax and duty are equivalent in this context). In this Annex, the term "electricity duty" is used (abbreviated to "ED" in table headings). The duty can be proportional to fixed charges and energy charges levied by the supplier, or it can be an amount per energy unit, which may depend on consumption as in a slab tariff.

Block 1: TransferCredit – Refer Table B.1.

Block 2: Electricity duty **–** The token structure of block 2 for Class 5, SubClass 10, tariff type 2 is given in Table B.19 and Table B.20.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table B.19 – Block 2 for class 5, SubClass 10, tariff type 2, sub-type 0 (TransferCredit + electricity duty)

Block sequence	Complete tariff	Tariff type	Tariff sub info	Tariff activation month	Data
2 bits	1 bit	4 bits	4 bits	6 bits	44 bits
Value: 0 or 1 See B.1.2	See B.1.3	0010b See B.1.7	0001b: Bit 0 set – Electricity duty See B.1.8	See B.1.5	See Table B.20
NOTE For clarity, the structure and contents of the data field are given in a separate table.					

Table B.20 – Block 2 for class 5, SubClass 10, tariff type 2, sub-type 0 (TransferCredit + electricity duty) – data field

Data						
ED on energy charges	ED on fixed charges	Number of ED slabs	ED fixed charge rate	ED energy rate 1	ED slab size 1	ED RES 1
3 bits	3 bits	3 bits	12 bits	12 bits	9 bits	2 bits
See B.4.2	See B.4.3	See B.4.4	See B.4.5	See B.4.5	See B.4.6	Reserved

B.4.2 Electricity duty on energy charges

The value of this parameter determines whether and how duty is to be levied on energy charges. Its interpretation is according to B.4.5.2.

NOTE If the tariff is based on apparent energy then the energy unit would be kVAh rather than kWh.

B.4.3 Electricity duty on fixed charges

The value of this parameter determines whether and how duty is to be levied on fixed charges. Its interpretation is according to B.4.5.2.

B.4.4 Number of electricity duty slabs

The slabs are defined as follows:

- 0 – flat, i.e. no slab value is required and none is sent in the token;
- 1 – two slabs, so one slab size value is to be sent in the token;
- 2 – three slabs, so two slab size values are to be sent in the token;
- 3 – four slabs, so three slab size values are to be sent in the token;
- 4 to 7 – reserved.

B.4.5 Electricity duty rate

B.4.5.1 General

Electricity duty rates are sent in 12-bit fields having a possible range of values of 0 to 4095.

B.4.5.2 Electricity duty amount scaling

The interpretation and value of the rate fields for both fixed and energy charges are according to the following list:

- 0 – duty is not levied on the charges;
- 1 – amount is in units of 0,01 minor currency units;

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

- 2 – amount is in units of 0,1 minor currency units;
- 3 – amount is in minor currency units;
- 4 – amount is in units of 10 minor currency units;
- 5 – amount is in units of 100 minor currency units;
- 6 – amount is in units of 1 000 minor currency units;
- 7 – duty is levied on energy charges as a percentage of charges.

B.4.5.3 Electricity duty rate – percentage

If the duty is levied as a percentage, the field is interpreted as follows:

- 0 to 2 000 – 0,00 % to 20,00 % in units of 0,01 %;
- 2 001 to 3 600 – 20,05 % to 100,00 % in units of 0,05 %;
- 3 601 to 4 000 – 100,1 % to 140,0 % in units of 0,1 %;
- 4 001 to 4 095 – reserved.

B.4.5.4 Electricity duty rate – amount

If the duty is levied as a currency amount, the field is interpreted as a monetary value according to the electricity duty amount scaling (see B.4.5.2).

If the duty is levied as a % of energy charge or % of fixed charge, the field interpreted as a % value according to electricity duty rate – percentage (see B.4.5.3).

B.4.6 Electricity duty slab size

The value is the size in kWh (or kVAh if the tariff is in kVAh) of the energy slab to which the electricity duty rate applies.

NOTE The starting point of each slab is the finishing point of the preceding one. For example, if the first slab size is 100 and the second is 250, then the first slab rate applies if consumption is from 0 to 100 kWh (or kVAh) and the second if consumption is >100 to 350 kWh (or kVAh).

The value is sent in 9 bits interpreted according to Table B.21.

Table B.21 – Electricity duty slab value encoding

Range in token	Steps in range	Minimum slab value	Maximum slab value	Slab value resolution
0 to 100	101	0	100	1
101 to 180	80	105	500	5
181 to 230	50	510	1000	10
231 to 250	20	1050	2000	50
251 to 330	80	2100	10000	100
331 to 410	80	10500	50000	500
411 to 460	50	51000	100000	1000
461 to 511	51	105000	355000	5000

Block 3: Electricity duty – The token structure of block 3 for class 5, SubClass 10, tariff type 2 is given in Table B.22.

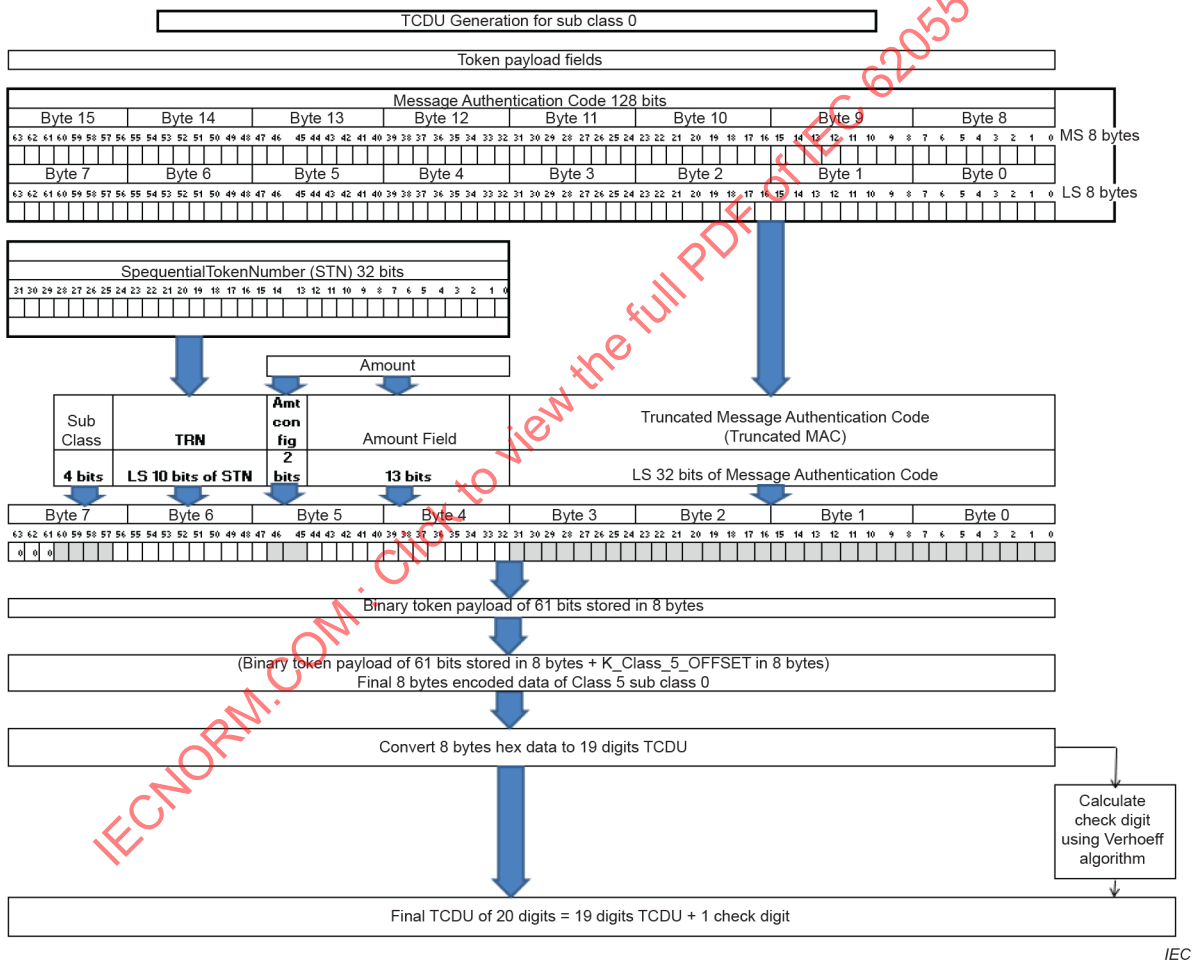
IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Table B.22 – Block 3 for class 5, SubClass 10, tariff type 2, sub-type 0 (TransferCredit + electricity duty)

	Data					
	ED energy rate 2	**ED slab size 2**	**ED energy rate 3**	**ED slab size 3**	**ED energy rate 4**	**ED RES 2**
2 bits	12 bits	9 bits	12 bits	9 bits	12 bits	5 bits
Value: 0 See B.1.2	See B.4.5	See B.4.6	See B.4.5	See B.4.6	See B.4.5	Reserved

B.5 SubClass 0 TCDU generation detailed process

The process for generating the TCDU for SubClass 0 is shown in Figure B.1.



IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Figure B.1 – TCDU generation process for SubClass 0

B.6 SubClass 8 TCDU generation detailed process

The process for generating the TCDU for SubClass 8 is shown in Figure B.2.

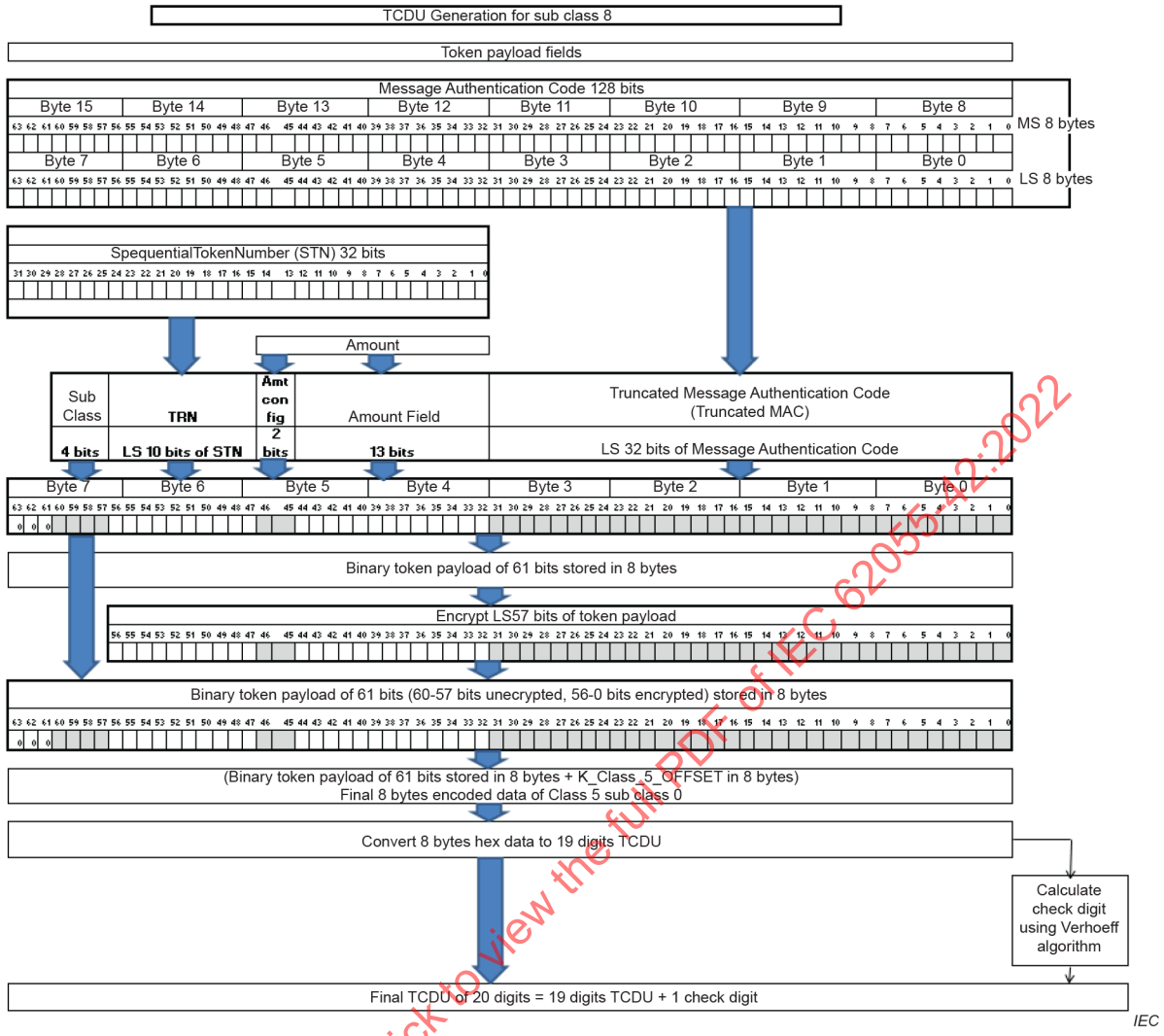


Figure B.2 – TCDU generation process for SubClass 8

B.7 SubClass 10 TCDU generation detailed process

The process for generating the TCDU for SubClass 10 is shown in Figure B.3.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

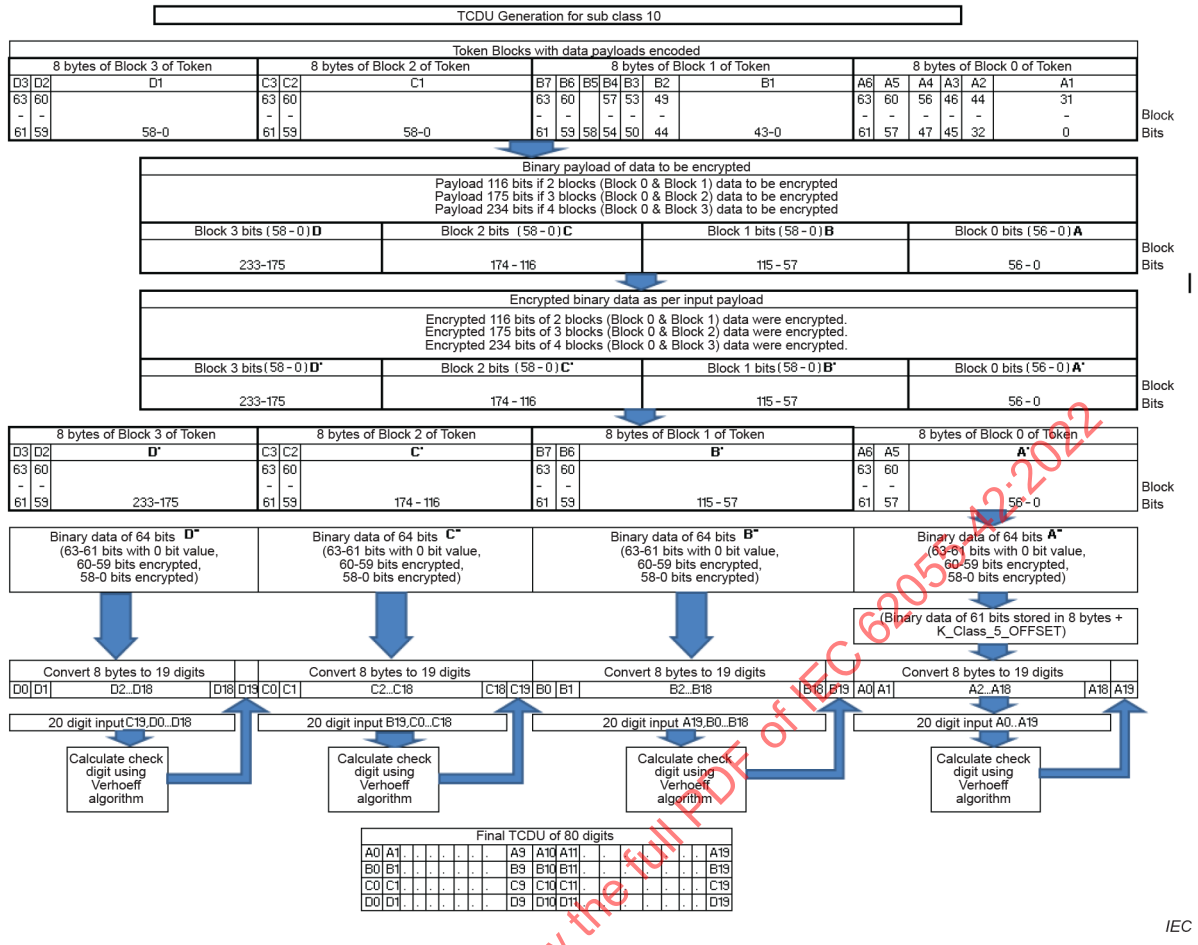


Figure B.3 – TCDU generation process for SubClass 10

B.8 SubClass 10 APDU extraction detailed process

The process for extracting the APDU for SubClass 10 is shown in Figure B.4.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

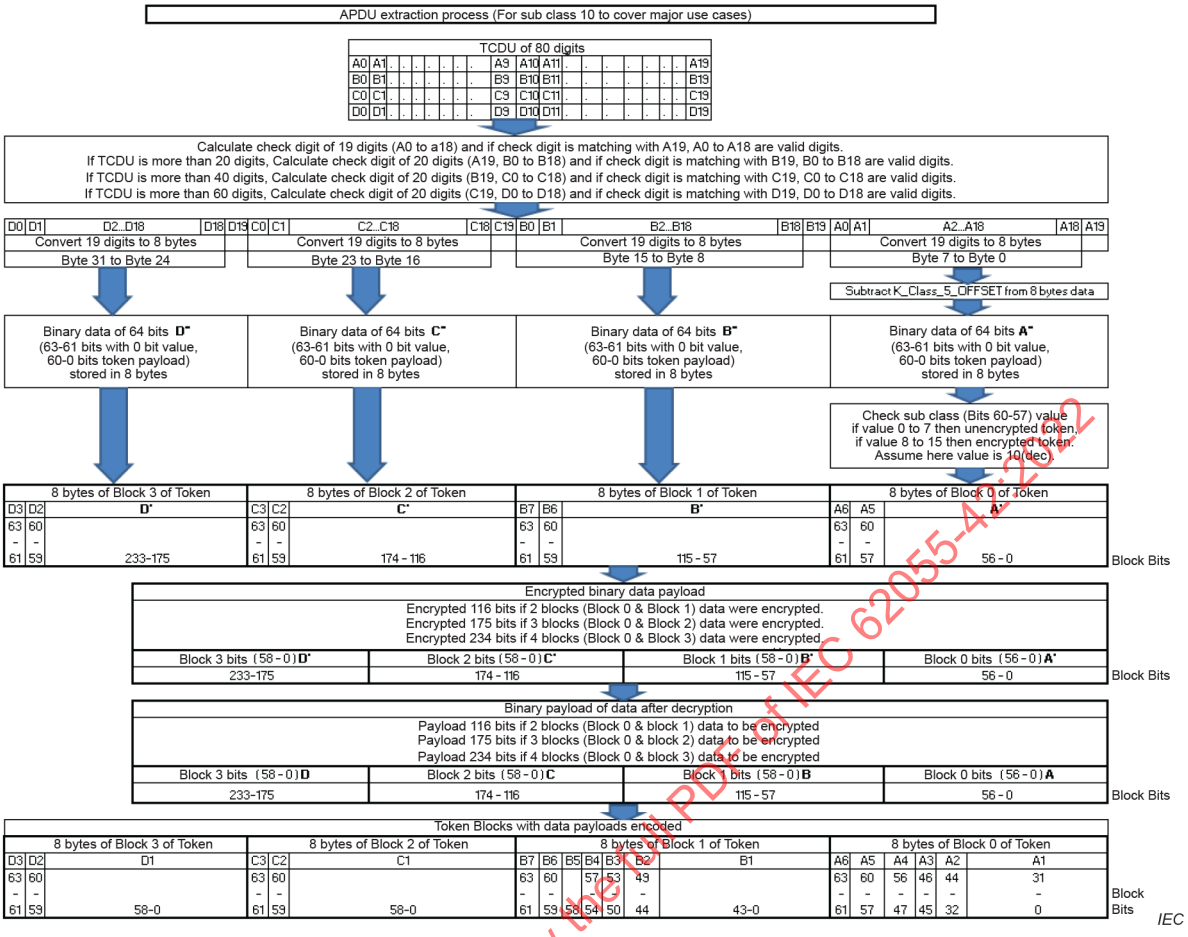


Figure B.4 – APDU extraction process for SubClass 10

The steps are as follows:

a) Based on the entered TCDU digits, block sequence numbers and verified check digits, the meter derives the number of blocks present in the TCDU;

b) In the first step, as the check digit/digits is/are already verified during the token entry process itself, check digit/digits is/are removed from the TCDU and the meter now has 19-digit block/blocks based on the entered token size;

c) In the second step, convert all 19-digit TCDU blocks to 8 bytes data present in the TCDU. For example: if the entered TCDU is of 20 digits, then MS 19 digits of the TCDU are to be converted to an 8-byte data block for block 0;

d) If the entered TCDU is 40 digits, then the two MS 19 digits of the two blocks of the TCDU are to be converted to two 8-byte data blocks for block 0 and block 1;

e) If the entered TCDU is 60 digits, then three MS 19 digits of the three blocks of the TCDU are to be converted to three 8-byte data blocks for block 0 to block 2;

f) If the entered TCDU is 80 digits, then four MS 19 digits of the four blocks of the TCDU are to be converted to four 8-byte data blocks for block 0 to block 3;

g) Subtract K_Class_5_OFFSET from 8-bytes data of block 0. There is no need to subtract K_Class_5_OFFSET from data blocks 1, 2 and 3 as only the first block is used to decide whether the TCDU is of Class 5 or not;

h) Extract the SubClass bits (Bits 60-57) from 8 bytes of data of block 0 . If the value of bit number 60 is 0 then the TCDU belongs to the unencrypted SubClass token and if value of bit 60 is 1 then the TCDU belongs to the encrypted SubClass token;

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

i) The value of Bits 60-57 is used to decide whether the TCDU comprises one block or whether it has multiple blocks. For example: if SubClass value is 8 or 9 then the TCDU will have only block 0;

j) If the SubClass value is 10 then the TCDU will have more blocks and based on block sequence value derived from TCDU, meter APDU extraction related application layer function will save data blocks in 8 bytes for each block as shown in Figure B.4;

k) For SubClass 10, the token can have more blocks and their validity is checked based on block sequence number of subsequent blocks;

l) Based on SubClass information and number of blocks in the token, the APDU extraction function extracts bits from each block and prepares the buffer for data to be decrypted. For example: If the token is of block 0 only then the number of data bits to be decrypted will be 57 bits;

m) If the token contains more than 1 block, then the data bits to be decrypted will be (57 bits) plus (59 bits of each additional block);

n) For 2 blocks, 3 blocks and 4 blocks encrypted tokens, the encrypted bits are 116 bits, 175 bits and 234 bits respectively;

o) After the decryption process, all decrypted bits are again copied to their respective data blocks;

p) Now blocks 0 to 3 have encoded data bits as per SubClass information and further information in the decrypted data of blocks 1 to 3;

q) Reconstruct STN from TSTN and check whether the token is a repeat, or whether STN is outside the sliding window (see 6.1.8);

r) Reconstruct the MAC and verify that the truncated 32 bits match those delivered in the token (see 6.1.13, 6.1.14, 6.1.15 and 6.1.16);

s) Apply the appropriate AuthenticationResult (see 7.1.3) and ValidationResult (see 7.1.4);

t) Carry out the functions and processes specified in the token APDU and apply the appropriate TokenResult (see 7.1.5);

u) Now APDU has all valid data to be saved and to be used as per parameter applicability to meter application layer process.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Bibliography

IEC 62055-51, *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, *Electricity metering – Payment systems – Part 52: Standard transfer specification (STS) – Physical layer protocol for a two-way virtual token carrier for direct local connection*

IEC 62056-4-7:2015, *Electricity metering data exchange – The DLMS/COSEM suite – Part 4-7: DLMS/COSEM transport layer for IP networks*

IEC 62056-6-1, *Electricity metering data exchange – The DLMS/COSEM suite – Part 6-1: Object Identification System (OBIS)*

IEC 62056-6-2, *Electricity metering data exchange – The DLMS/COSEM suite – Part 6-2: COSEM interface classes*

IEC 62056-21, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 21: Direct local data exchange*

IEC 62056-42, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 42: Physical layer services and procedures for connection-oriented asynchronous data exchange*

IEC 62056-46, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 46: Data link layer using HDLC protocol*

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

SOMMAIRE

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

COMPTAGE DE L'ÉLECTRICITÉ – SYSTÈMES DE PAIEMENT –

Partie 42: Numéros de référence des transactions (TRN)

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.
9) L'attention est attirée sur le fait que certains des éléments de la présente publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et averti de leur existence.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

L'IEC 62055-42 a été établie par le comité d'études 13 de l'IEC: Comptage et pilotage de l'énergie électrique. Il s'agit d'une Norme internationale.

Le texte de cette Norme internationale est issu des documents suivants:

Projet	Rapport de vote
13/1843/CDV	13/1860/RVC

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

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2, il a été développé selon les Directives ISO/IEC, Partie 1 et les Directives ISO/IEC, Supplément IEC, disponibles sous www.iec.ch/members_experts/refdocs. Les principaux types de documents développés par l'IEC sont décrits plus en détail sous www.iec.ch/standardsdev/publications.

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

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous webstore.iec.ch dans les données relatives au document recherché. A cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de ce document indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

INTRODUCTION

La série IEC 62055 identifie et applique le concept d'interopérabilité entre les couches dans les domaines du comptage intelligent et des réseaux intelligents.

Elle assure également l'interopérabilité des éléments du système au-dessus de la couche sémantique afin d'intégrer les couches d'interopérabilité des fonctions et processus métier au sein d'un système de comptage de l'électricité, en assurant ainsi la compatibilité globale à tous ces niveaux.

Le présent document est fondé sur les principes des normes IEC 62055 et établit les règles qui assureront la cohérence des futures extensions, en fournissant ainsi aux entreprises de distribution un vocabulaire commun pour formuler des exigences dans le cadre d'appels d'offres, et aux vendeurs une base de connaissances unifiée pour l'interprétation des exigences relatives aux appels d'offres.

Le présent document fait partie de la série IEC 62055 et partage plusieurs références avec l'IEC 62055-41, les deux normes représentant les jetons TransferCredit en utilisant des supports de jetons à 20 chiffres. En revanche, l'IEC 62055-41 et l'IEC 62055-42 présentent de nettes différences pour ce qui est du codage, du mécanisme de sécurité et des cas d'utilisation prévue. L'IEC 62055-41 porte essentiellement sur les systèmes hors ligne, alors que l'IEC 62055-42 est principalement destinée aux systèmes en ligne dans lesquels le support de jeton décimal fait office de mécanisme de secours pour la vente lorsque les compteurs sont hors ligne par intermittence.

L'IEC TC13 a spécifiquement développé la série IEC 62055 pour les systèmes de comptage de l'électricité, mais cette série peut également s'appliquer à d'autres services d'utilité publique tels que l'eau et le gaz.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

COMPTAGE DE L'ÉLECTRICITÉ – SYSTÈMES DE PAIEMENT –

Partie 42: Numéros de référence des transactions (TRN)

1 Domaine d'application

Le présent document spécifie un mécanisme de génération de jetons ainsi que la structure de jetons associée à la fonctionnalité de prépaiement intelligent, destinée aux marchés où les systèmes conformes à l'IEC 62055-41 ne sont pas utilisés et lorsque les exigences nationales ou spécifiques à un projet imposent l'utilisation d'un autre mécanisme de sécurité. Le présent document spécifie la structure des jetons, leur authentification, un mécanisme antirediffusion, le modèle d'exploitation des jetons et le protocole associé.

Le présent document est informé par les services de gestion de clés de la STS Association ainsi que par les mécanismes de gestion de clés utilisés dans le modèle de sécurité DLMS/COSEM spécifié dans l'IEC 62056-6-2. Une référence est faite aux normes internationales relatives aux jetons STS (IEC 62055-41, IEC 62055-51 et IEC 62055-52) pour les systèmes de comptage à paiement, et l'interconnexion a été envisagée le cas échéant en ce qui concerne les plages des supports de jetons dans le domaine décimal. Les jetons conformes à l'IEC 62055-41 et ceux décrits dans le présent document ne sont pas interopérables, mais leurs domaines sont conçus pour être mutuellement exclusifs afin d'assurer que les deux types de jetons n'interfèrent pas l'un avec l'autre.

Le traitement et la fonctionnalité des applications de comptage ainsi que les commandes et attributs des interfaces HAN et WAN ne relèvent pas du domaine d'application du présent document. Une référence est toutefois faite à d'autres normes traitant de ces aspects.

Le mécanisme d'audit et de récupération des données du compteur relatives à la tarification, des relevés de compteurs, des données sur les profils et autres informations métrologiques légales ne relève pas du domaine d'application du présent document. Il est cependant défini dans le cadre de toute solution de comptage globale. Il est permis de définir de telles interfaces pour la récupération des données d'un compteur en utilisant des protocoles adaptés, tels que DLMS/COSEM défini dans la série IEC 62056.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. 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 60050-300:2001, *Vocabulaire Electrotechnique International (IEV) – Partie 300: Mesures et appareils de mesure électriques et électroniques – Partie 311: Termes généraux concernant les mesures – Partie 312: Termes généraux concernant les mesures électriques – Partie 313: Types d'appareils électriques de mesure – Partie 314: Termes spécifiques selon le type d'appareil*
IEC 60050-300:2001/AMD1:2015
IEC 60050-300:2001/AMD2:2016
IEC 60050-300:2001/AMD3:2017
IEC 60050-300:2001/AMD4:2020

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

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

IEC 62055-31:2005, *Equipements de comptage de l'électricité – Systèmes à paiement – Partie 31: Exigences particulières – Compteurs statiques à paiement d'énergie active (classes 1 et 2)*

IEC 62055-41:2018, *Partie 41: Spécification de transfert normalisé (STS) – Protocole de couche application pour les systèmes de supports de jeton unidirectionnel*

IEC 62056-5-3:2017, *Echange des données de comptage de l'électricité – La suite DLMS/COSEM – Partie 5-3: Couche application DLMS/COSEM*

IEEE EUI-64, https://standards.ieee.org/develop/regauth/tut/eui64.pdf

Verhoeff, J., 1975, *Error Detecting Decimal Codes* (Mathematical Centre Tracts, 29)

NIST SP 800-38D: 2007, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

3 Termes, définitions, termes abrégés et notation

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans l'IEC 60050-300:2001, l'IEC TR 62051:1999, l'IEC 62055-31:2005, l'IEC 62055-41:2018 et l'IEC TR 62055-21:2005, ainsi que les suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse http://www.electropedia.org/
- ISO Online browsing platform: disponible à l'adresse http://www.iso.org/obp

NOTE Lorsqu'il existe une différence entre les définitions du présent document et celles contenues dans d'autres normes IEC de référence, les définitions du présent document prévalent.

3.1.1
spécification d'accompagnement
ensemble d'exigences et de choix opérés à partir d'une norme de base, spécifique à une région, un consortium ou un fabricant, afin de faciliter un ensemble particulier de cas d'utilisation

3.1.2
compteur
instrument de mesure; synonyme de "compteur à paiement" et de "décodeur" lorsque le décodeur est une sous-partie d'une installation à dispositifs multiples

3.1.3
POS
point de vente; entité qui peut créer et transférer des jetons, synonyme de "SIC" (système d'information des consommateurs), "SIG" (système d'informations de gestion) et "TSP" (terminal de saisie portable)

3.1.4
super jeton
groupe de chiffres issus de plusieurs blocs de données de l'APDU, qui doivent être saisis les uns à la suite des autres pour former un message de plus grande taille

3.1.5
entreprise de distribution
détaillant ou fournisseur d'un service de distribution d'énergie ou d'eau

Note 1 à l'article: 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 un fournisseur de service tiers.

3.2 Termes abrégés

AES	Advanced Encryption Standard (norme de chiffrement avancé)
AAD	AdditionalAuthenticatedData (données authentifiées supplémentaires)
AMT	AMounT (montant)
APDU	ApplicationProtocolDataUnit (unité de données de protocole de couche application)
SIC	Système d'information des consommateurs
COSEM	COmpanion Specification for Electricity Metering (spécification d'accompagnement pour le comptage de l'électricité)
DLMS	Device Language Message Specification (spécification de message de langage de dispositif)
EUI	Extended Unique Identifier (identifiant unique étendu)
GMAC	Galois Message Authentication Code (code d'authentification du message de Galois)
HAN	Home Area Network (réseau local domestique)
HES	Head End System (système tête de réseau)
HHU	Hand Held Unit (unité portable)
IHM	Interface Homme-Machine
ID	IDentifiant
IHD	In-Home Display (affichage au domicile)
IEEE	Institute of Electrical and Electronics Engineers (Institut des ingénieurs électriciens et électroniciens)
KMS	Key Management System (système de gestion de clés)
LCS	Load Control Switch (interrupteur de la charge)
LSB	Least Significant Byte (octet de poids faible)
MAC	MessageAuthenticationCode (code d'authentification de message)
SIG	Système d'informations de gestion
MOD	Opération modulo
MS	Most Significant (de poids fort)
MSB	Most Significant Byte (octet de poids fort)
NIST	National Institute of Standards and Technology (Institut national américain des normes et des technologies)
NSP	National Service Provider (fournisseur de services national)
PIN	Personal Identification Number (numéro d'identification personnel)
POS	Point Of Sale (point de vente)
RANDz	Reasonable And Non-Discriminatory with zero royalty (raisonnable et non discriminatoire avec zéro redevance)
RES	Réservé
SE	Smart Energy (énergie intelligente)
SMS	Short Message Service (service de messages courts)

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

STN	SequentialTokenNumber (numéro de jeton séquentiel)
STS	Standard Transfer Specification (spécification de transfert normalisé)
TC	Technical Committee (comité d'études)
TCDU	TokenCarierDataUnit (unité de données de support de jeton)
TMAC	Truncated Message Authentication Code (code d'authentification de message tronqué)
TOU	Time Of Use (période d'utilisation)
TSTN	TruncatedSequentialTokenNumber (numéro de jeton séquentiel tronqué)
WAN	Wide Area Network (réseau étendu)
WG	Working Group (groupe de travail)

3.3 Notation

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.

NOTE La notation utilisée pour le nommage d'objets a été alignée sur ladite "notation-chameau" utilisée dans les normes du modèle d'information commun (CIM, Common Information Model) élaborées par l'IEC/TC57, afin de faciliter l'harmonisation et l'intégration futures des normes des systèmes de paiement avec les normes CIM.

Le symbole "||"inséré entre deux expressions implique que les valeurs résultantes des deux expressions sont concaténées en une seule chaîne sans séparateurs.

4 Conventions de numérotation utilisées dans le présent document

Les nombres sont au format décimal, sauf indication contraire. Les valeurs des chiffres décimaux se situent dans la plage 0 à 9.

Les valeurs des chiffres binaires se situent dans la plage 0 à 1. Les nombres binaires sont précédés du suffixe "b". La représentation en chaînes binaires est telle que le bit de poids faible est à droite de la représentation et le bit de poids fort à gauche de la chaîne. La numérotation des positions des bits commence par la position du bit 0, qui correspond au bit de poids faible du nombre binaire.

Les valeurs des chiffres hexadécimaux se situent dans les plages 0 à 9 et A à F et sont indiquées par le préfixe "0x" ou le suffixe "hex".

5 Modèle de référence d'un compteur intelligent

5.1 Diagramme fonctionnel générique de référence

Dans le cas général, tout compteur conçu pour être conforme au présent document peut contenir tout ou partie des fonctions spécifiées à la Figure 1 et un minimum de 1 type de crédit et 1 type de charge (il est admis d'en utiliser davantage). Le schéma de la Figure 1 est fourni ici à titre de référence et de contexte, mais n'entre pas dans le domaine d'application du présent document.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

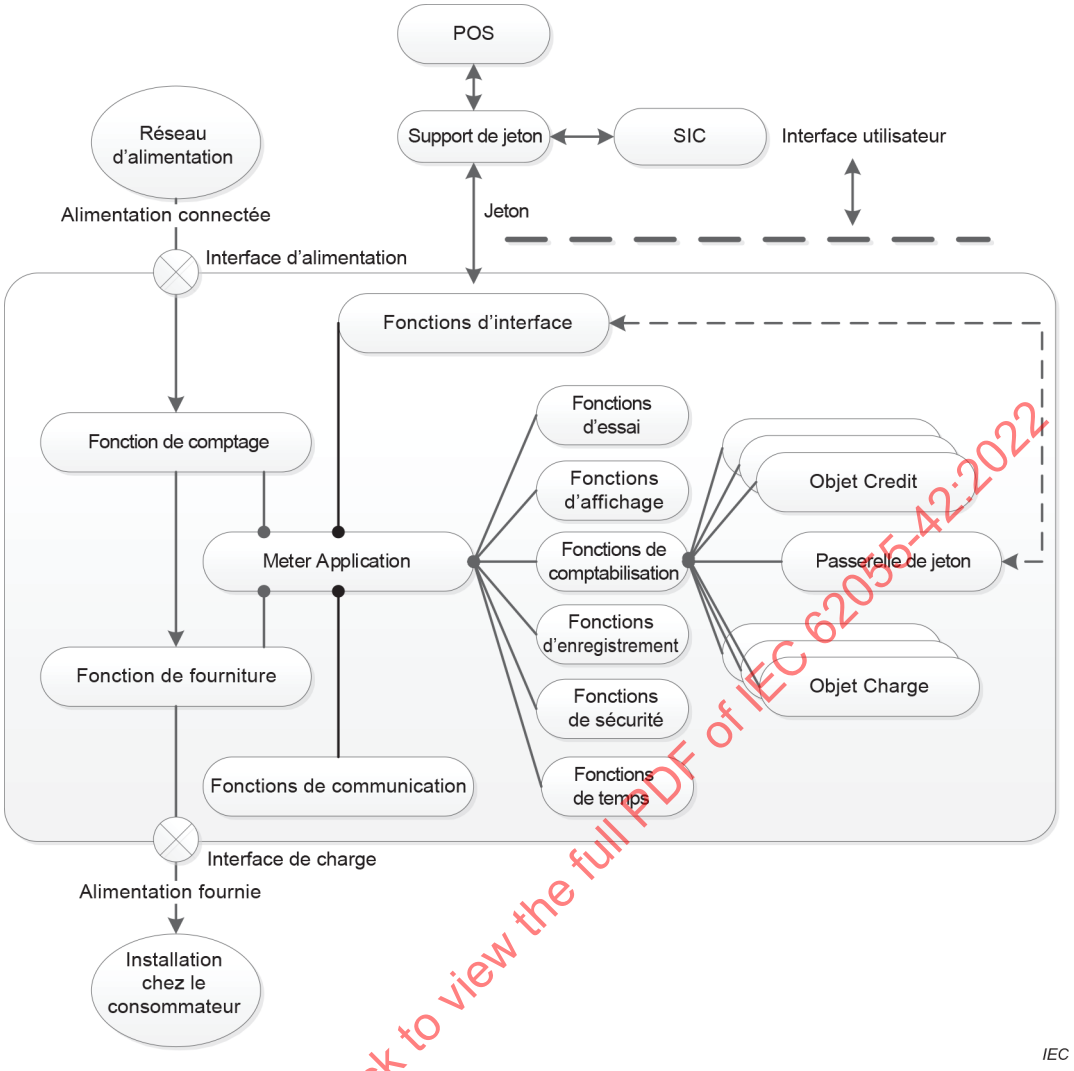


Figure 1 – Organigramme fonctionnel d'un compteur à paiement générique

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Le présent document porte principalement sur la structure, la génération, le transport et la sécurité du jeton lui-même, comme le montre la Figure 1. La génération des jetons, dans le cas des jetons reçus par un compteur, a lieu avec une forme de système logiciel à l'extérieur du domaine du compteur en utilisant des informations qui sont spécifiques à un compteur et au tiers qui génère les jetons. Dans le cas des jetons émis par un compteur, la génération a lieu à l'intérieur du compteur en utilisant les informations se rapportant au compteur spécifique et au destinataire prévu de ces jetons. Ces informations contiennent, entre autres, les informations d'identification du compteur et le matériau de sécurité tel que les clés cryptographiques et les clés d'identification.

Le modèle représente un compteur à paiement type, équipé de toutes les fonctions indiquées ci-dessus; il est toutefois admis que d'autres architectures à dispositifs multiples soient compatibles avec le modèle générique. Par exemple, la fourniture du jeton pourrait avoir lieu par le biais d'un dispositif proxy.

Dans cette architecture, il est également possible de délivrer un jeton par le biais de l'interface HAN ou WAN, ou d'une interface homme-machine (IHM) distincte.

Etant donné la diversité des cas d'utilisation exigés pour le comptage intelligent, il peut s'avérer nécessaire de transporter/tunnelliser un jeton par le biais de protocoles multiples. Ceci implique qu'un protocole de transport commun n'est pas pratique et n'est pas nécessaire pour la présente norme.

La seule exigence relative au transport du jeton est de définir le format des données et la charge utile du jeton, en s'assurant que le format de ce dernier est simple et suffisamment flexible pour être intégré sous la forme d'un objet tunnellisé dans tout protocole de transport (du point de vue du jeton).

5.2 Modèle de référence de protocole de transfert de jetons

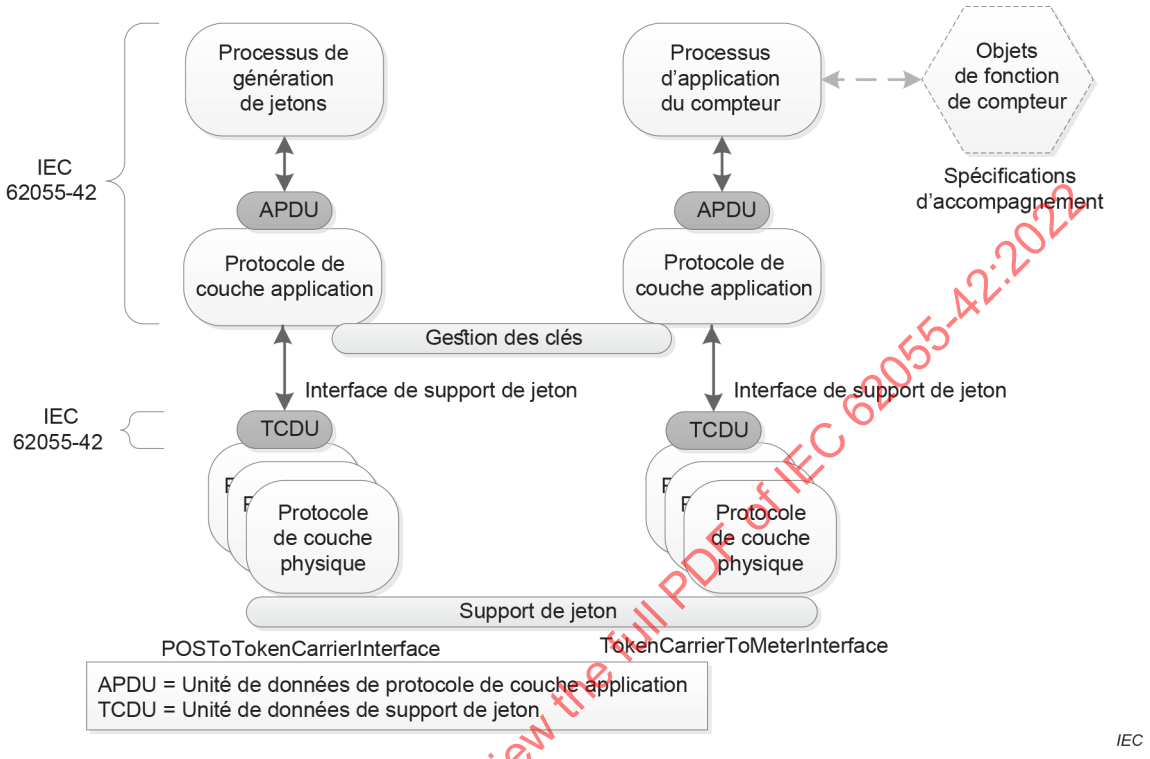


Figure 2 – Modèle de référence sous forme de pile protocolaire OSI réduite à 2 couches

La Figure 2 représente le système de génération de jetons et l'interface de compteur en utilisant une représentation protocolaire décrivant un système qui génère un jeton en utilisant plusieurs éléments de données concaténés spécifiques pour obtenir une charge utile binaire, qui est facultativement chiffrée (en fonction de SubClass), en créant une ApplicationProtocolDataUnit (APDU). La charge utile binaire est ensuite convertie en nombre décimal qui est appelé TokenCarrierDataUnit (TCDU) et cette TCDU peut être transportée vers un compteur d'une manière ou d'une autre, par le biais d'une couche de protocole physique (ce transfert peut être effectué par l'intermédiaire d'un support de jeton imprimé sur papier, par liaison radio ou tout autre mécanisme). Lorsqu'une interaction humaine est exigée (par exemple en cas d'impression, de saisie au clavier ou de communication orale), la TCDU doit être sous forme décimale mais elle peut être représentée dans un autre format (par exemple pour un transport par liaison radio). Après le transport de la TCDU jusqu'à l'interface du compteur, la couche de protocole application du compteur est tenue de convertir et de déchiffrer la TCDU et de délivrer la charge utile à la fonction appropriée dans le processus d'application du compteur à paiement.

La TCDU peut être délivrée au compteur de diverses manières, comprenant entre autres:

- à distance par une interface de communication ("par liaison radio");
- localement par un support de jeton physique;
- processus de saisie manuelle initié par l'utilisateur.

En plus d'une TCDU créée pour la fourniture à un compteur, le présent document permet également au compteur de créer une TCDU en vue d'une fourniture et d'un traitement par un autre système logiciel externe (voir 6.2.5.4).

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

5.3 Flux de données du POSApplicationProcess vers le TokenCarrier

La Figure 3 représente un processus générique qui permet de créer une TCDU à partir d'une APDU, en concaténant des éléments de données binaires pour construire une APDU puis en exécutant un certain nombre d'opérations obligatoires et facultatives pour convertir l'APDU en une TDCU décimale. La TCDU est ensuite utilisée comme charge utile codée pour l'envoyer vers un compteur à paiement sur une diversité de supports physiques et de transport.

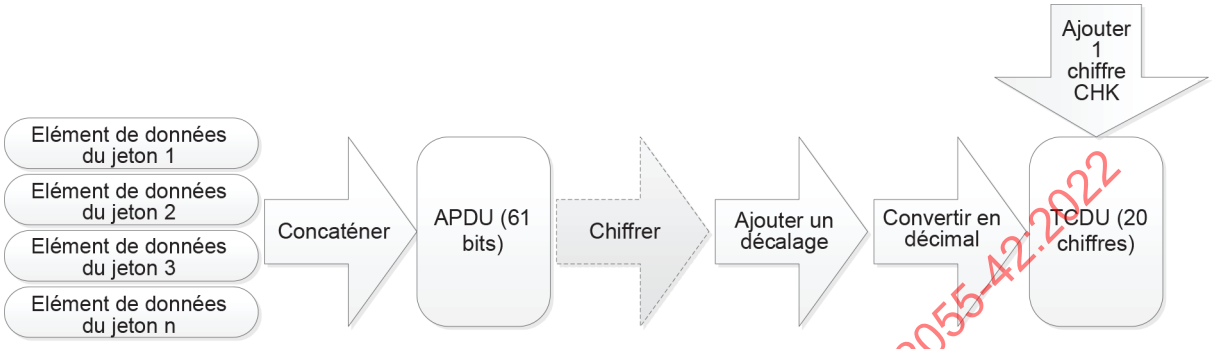


Figure 3 – Modèle générique du POSApplicationProcess vers le TokenCarrier

5.4 Flux de données du TokenCarrier vers le MeterApplicationProcess

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

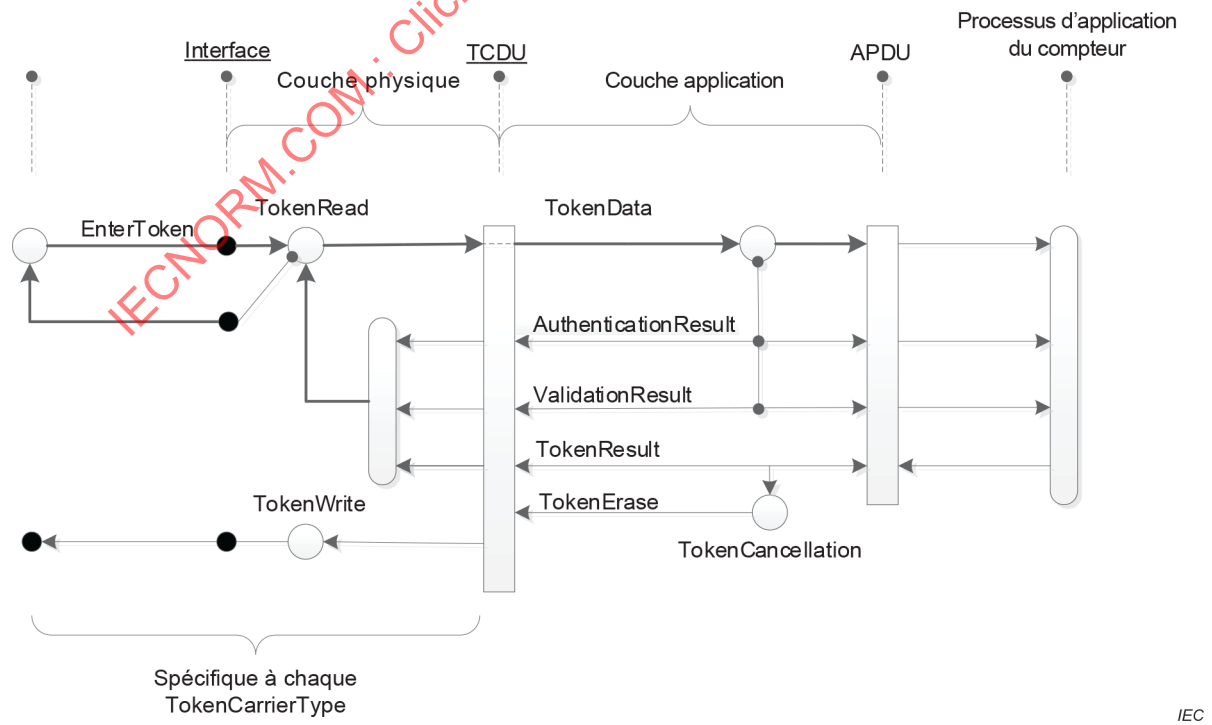


Figure 4 – Flux de données du TokenCarrier vers le MeterApplicationProcess

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

5.5 MeterFunctionObjects / spécifications d'accompagnement

Comme représenté à la Figure 2, la TokenCarrierToMeterInterface, qui inclut également le TokenCarrier, est traitée dans le présent document. Les MeterFunctionObjects restants qui figurent dans le diagramme sont définis dans des spécifications d'accompagnement et ne sont pas normatifs dans le présent document. (Voir 3.1.1.)

6 Protocole de couche application POSToTokenCarrierInterface

6.1 APDU: ApplicationProtocolDataUnit

6.1.1 Eléments de données dans l'APDU

La charge utile de Token dans l'APDU est une charge utile binaire (61 bits pour SingleTokenPayload) comportant un certain nombre d'éléments de données plus petits indiqués en 6.3.

Pour construire une APDU complète, des éléments de données supplémentaires sont exigés, tels que SubClass, Truncated SequentialTokenNumber (TSTN), Data et TruncatedMessageAuthenticationCode (TMAC).

Le Tableau 1 détaille les éléments de données Basic et Derived pour la construction de l'APDU.

Tableau 1 – Eléments Basic et Derived pour la construction de l'APDU et de la TCDU

Nom de l'élément de données	Contexte	Format de l'élément de données	Taille	Plage	Basic/ Derived	Référence
AdditionalAuthenticationData (AAD)	Octets de données supplémentaires qui peuvent être placés dans la chaîne de calcul d'un MAC.	Binaire	Variable	Variable	Derived	6.1.14
Data	Les données dans l'APDU dépendent du type SubClass.	Binaire	Variable	Toutes les valeurs possibles basées sur la taille des données	Données Basic ou Derived selon le type SubClass	
FunctionIndex	Paramètre utilisé pour appliquer l'acceptation de jeton à l'extrémité réceptrice dans un ordre particulier.	Binaire	32 bits	Toutes les valeurs possibles	Basic	6.1.9
InitializationVector (IV)	Partie de l'AAD exigée pour l'algorithme GMAC.	Binaire	96 bits	Valeur fixe de 0x00000000 concaténée avec le SupplierID de 64 bits	Derived	Figure 7

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Nom de l'élément de données	Contexte	Format de l'élément de données	Taille	Plage	Basic/ Derived	Référence
MessageAuthenticationCode (MAC)	Code d'authentification de message généré en utilisant la TokenPayload et d'autres éléments de données. Ce MAC est ensuite tronqué pour former le TruncatedMAC et le TMAC est ajouté à la TokenPayload en tant que LSB pour former l'APDU.	Binaire	128 bits	Toutes les valeurs possibles	Derived	6.1.13
MessageIdentifier	Octet de données utilisé pour préparer l'AAD et constitué de multiples éléments de données en fonction du type SubClass.	Binaire	Variable	Variable	Derived	6.1.5
MeterID	Identification individuelle unique d'un compteur spécifique.	Binaire	64 bits		Basic	6.1.3
Offset	Valeur numérique constante utilisée pour la production d'une TCDU afin de décaler le jeton numérique vers une valeur compatible avec les jetons de classe 0, 1, 2, 3, 4 ou 6.	Binaire ou décimal selon le contexte d'utilisation	63 bits ou 19 chiffres	Valeur spécifique détaillée dans le Tableau 9		6.1.17
SequentialTokenNumber (STN)	Numéro de jeton unique à chaque compteur, utilisé en totalité pour créer un MAC pour une APDU, dont seuls les 10 bits LS sont inclus dans l'APDU et sont désignés TruncatedSequential TokenNumber (TSTN).	Binaire	32 bits	Toutes les valeurs possibles	Basic	6.1.6
SingleTokenPayload	Données réelles du jeton qui sont à transférer vers le compteur avant traitement, incluant le MAC tronqué et/ou le chiffrement, avec 3 bits MS supplémentaires 000 b pour le bourrage jusqu'à 64 bits.	Binaire	61 bits	Toutes les valeurs possibles	Derived	6.1.11

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Nom de l'élément de données	Contexte	Format de l'élément de données	Taille	Plage	Basic/ Derived	Référence
SuperTokenPayload	Charge utile considérée provenant de plusieurs jetons lorsqu'une série de jetons est présentée au compteur de façon ordonnée. Cette charge utile est plus grande que les 61 bits car elle doit incorporer la charge utile initiale de 61 bits du premier bloc de données de la série (avec 3 bits MS supplémentaires avec 000b pour le bourrage jusqu'à 64 bits), puis toutes les charges utiles binaires des blocs de données suivants de l'ensemble. Pour chaque bloc de données à partir du bloc 2 de l'ensemble, 3 bits supplémentaires de valeur 000b doivent être inclus pour le bourrage afin d'obtenir des charges utiles équivalentes de 64 bits.	Binaire	122 bits Ou 183 bits Ou 244 bits	3 valeurs discrètes selon si la longueur du SuperToken est de 2 ou 3 ou 4 blocs.	Derived	6.1.12
SubClass	Définit le type de jeton et si le jeton est chiffré ou non. Cette information n'est jamais chiffrée dans le jeton.	Binaire	4 bits	Toutes les valeurs possibles	Basic	6.2.3 et 6.2.4
SuperTokenBlockToFollow	Identification du numéro du bloc dans la chaîne de blocs de données dans le cas du jeton SubClass 10. Cette valeur représente le numéro des blocs à respecter.	Binaire	2 bits	Toutes les valeurs possibles (00b, 01b, 10b) sauf 11b		Voir B.1.2
SupplierID	Numéro d'identification spécifique au tiers qui crée le jeton.	Binaire	64 bits	Toutes les valeurs possibles	Basic	6.1.2
TruncatedSequnetialTokenNumber (TSTN)	Comme la taille des données du STN est de 32 bits, ses 10 bits LS, désignés TruncatedSequential TokenNumber (TSTN), sont utilisés dans les données de la charge utile de l'APDU.	Binaire	10 bits	Toutes les valeurs possibles	Derived	6.1.7
TruncatedMAC (TMAC)	32 bits LS du MAC	Binaire	32 bits	Toutes les valeurs possibles	Derived	6.1.13

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Nom de l'élément de données	Contexte	Format de l'élément de données	Taille	Plage	Basic/ Derived	Référence
TokenOriginationID	Indication du sens du jeton – émis ou reçu par un compteur.	Binaire	8 bits	0x01, 0x02	Basic	6.1.4

6.1.2 SupplierID

Le SupplierID identifie le créateur du jeton de façon unique, qui peut être une entreprise de distribution, un réseau de vente ou un autre fournisseur de services tiers. Le SupplierID peut être une valeur spécifique à un projet affectée à un compteur spécifique et doit être globalement unique. En général, il pourrait s'agir d'un titre système d'un client, tel que défini dans DLMS/COSEM (voir l'IEC 62056-5-3:2017, 4.1.3.4) ou un IEEE EUI-64 ou un autre identifiant à 64 bits choisi pour un projet et géré par une autorité d'enregistrement officielle ou un autre tiers afin d'assurer son unicité.

6.1.3 MeterID

Le MeterID peut être une valeur spécifique à un projet affectée à un compteur spécifique et doit être globalement unique. En général, il pourrait s'agit d'un titre système d'un compteur (serveur), tel que défini dans DLMS/COSEM (voir l'IEC 62056-5-3:2017, 4.1.3.4) ou un IEEE EUI-64 ou un autre identifiant à 64 bits choisi pour un projet et géré par une autorité d'enregistrement officielle ou un autre tiers afin d'assurer son unicité.

6.1.4 TokenOriginationID

Ce champ indique si le jeton a été généré à partir du compteur ou du HES et s'il est inclus dans le calcul du MAC. Les valeurs affectées sont:

- 0x01 pour les instructions (à destination du compteur);
- 0x02 pour les réponses (en provenance du compteur);
- 0x03….0xFF réservées pour une affectation future.

6.1.5 MessageIdentifier

Il s'agit d'un tableau de données en octets qui est utilisé pour la dérivation de l'AAD. Les informations de dérivation des données du MessageIdentifier pour les sous-classes chiffrées et non chiffrées sont indiquées dans le Tableau 2.

Tableau 2 – Détail du MessageIdentifier par sous-classe et classe fonctionnelle SubClass

SubClass	Description	MessageIdentifier	Classe fonctionnelle SubClass	Notes
Sous-classes non chiffrées				
0	TransferCredit non chiffré	SupplierID ||MeterID ||TokenOriginationID (0x01) ||STN (compteur de jetons non-SpecialFunction généré par le fournisseur) ||FunctionIndex (facultatif)	1	FunctionIndex reste à 0 car aucune SubClass de jeton "TransferCredit + Function" n'est présente.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

SubClass	Description	MessageIdentifier	Classe fonctionnelle SubClass	Notes
1	Jeton SpecialFunction	SupplierID ||MeterID ||TokenOriginationID (0x01) ||STN (compteur de jetons SpecialFunction généré par le fournisseur)	3	
2 – 7	RESERVEES			
Sous-classes chiffrées				
8	TransferCredit	SupplierID ||MeterID ||TokenOriginationID (0x01) ||STN (compteur de jetons non-SpecialFunction généré par le fournisseur) ||FunctionIndex	1	FunctionIndex est la valeur utilisée dans le jeton "TransferCredit + Function" précédemment émis.
9	Jeton SpecialFunction	SupplierID ||MeterID ||TokenOriginationID (0x01) ||STN (compteur de jetons SpecialFunction généré par le fournisseur)	3	Chaque fois qu'un jeton SpecialFunction est accepté par un compteur, tous les jetons SpecialFunction précédents sont rejetés s'ils sont saisis par la suite.
10	TransferCredit + Function	SupplierID ||MeterID ||TokenOriginationID (0x01) ||SequentialTokenNumber (compteur de jetons non-SpecialFunction généré par le fournisseur) ||FunctionIndex	1	FunctionIndex = (dernier FunctionIndex + 1)
11	Jeton MeterGenerated	SupplierID ||MeterID ||TokenOriginationID (0x02) ||SequentialTokenNumber (compteur de jetons MeterGenerated)	Aucune règle de groupe SubClass exigée pour que le jeton soit accepté par le HeadEndSystem.	
12 – 15	RESERVEES			

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.1.6 SequentialTokenNumber (STN)

Cette valeur est un nombre à 32 bits qui est généré et géré par le HES (ou par le compteur en cas de jetons générés par compteur). Il s'agit d'un compteur croissant qui incrémente à chaque jeton généré pour ou par le compteur afin d'empêcher la rediffusion de jetons. Chaque jeton généré pour ou par un compteur particulier doit avoir un STN différent jusqu'au point de passage à zéro du STN. Cependant, il est présumé qu'avec des schémas TransferCredit normaux, la périodicité du passage à zéro est très longue (nettement supérieure à la durée de vie d'un compteur). Le STN doit être tronqué et les 10 bits de poids faible doivent être utilisés dans la charge utile du jeton et doivent être appelés TSTN (voir 6.1.7); cependant, la valeur complète de 32 bits du STN, qui est reconstruite par le compteur, doit être utilisée pour calculer le MAC pour le jeton.

La génération d'un jeton pour ou par tout compteur ne doit pas affecter la séquence d'instances de STN générées pour ou par tout autre compteur. Le premier jeton généré pour toute SubClass doit avoir une valeur de STN de 0x00000001.

6.1.7 TruncatedSequentialTokenNumber (TSTN)

Le STN complet à 32 bits n'est pas transféré afin d'économiser l'espace dans le jeton. Ses 22 bits MS changent très peu souvent et peuvent être déduits de manière fiable à partir des 10 bits LS d'un jeton; ces 10 bits LS composent le TSTN.

Il existe différentes manières de déduire le STN complet sur la base du TSTN, l'une d'entre elles utilisant le principe d'une fenêtre glissante au moyen de laquelle le compteur (ou le HES dans le cas d'un jeton généré par compteur) ne doit jamais accepter un jeton ayant un TSTN qui soit supérieur, selon une valeur au moins égale à X, ou inférieur, selon une valeur au moins égale à Y, au dernier jeton accepté, la somme non signée de X et Y n'étant pas supérieure à 256. La raison de ceci est d'empêcher le passage à zéro non observé du champ de 10 bits du STN (c'est-à-dire le TSTN). Comme les jetons plus anciens sont probablement présentés au compteur de manière plus désordonnée que les jetons plus récents, il est recommandé de biaiser la fenêtre glissante de l'acceptation des jetons vers le passé, ce qui signifie que Y sera plus grand que X.

6.1.8 Déduction de la partie MS du STN et validation du TSTN

Le STN, N, est validé en l'associant aux précédentes valeurs de STN acceptées par le compteur (ou le HES dans le cas d'un jeton généré par compteur) pour les jetons de la même SubClass.

Il est nécessaire que le compteur (ou le HES en cas de jeton généré par compteur) utilise les données suivantes pour déduire et valider le STN pour chaque SubClass:

a) M_N est la valeur maximale du N du jeton déjà accepté (par le compteur ou le HES selon la SubClass), M ayant une valeur initiale de 0 (sauf indication contraire dans une spécification d'accompagnement);

b) L_N est la valeur basse acceptable de N qui est valide, L_N ayant une valeur initiale de 1 pour toutes les sous-classes;

c) U_N est la valeur haute acceptable de N qui est valide (valeur initiale de 128 pour la SubClass 0, 128 valeur commune pour la SubClass 8 et la SubClass 10, alors que la valeur initiale est 512 pour la SubClass 9 dérivée du Tableau 3;

d) $LS(N)$ (c'est-à-dire le TSTN) est la partie de poids faible de N, qui est disponible dans la charge utile du jeton;

e) R est le nombre de valeurs possibles du TSTN. Par exemple: TSTN a un champ de 10 bits, R a une valeur de 1 024 et les valeurs possibles du TSTN sont comprises entre 0 et 1 023;

f) $MS(N)$ est inclus en tant qu'information d'entrée dans le MAC et peut être déduit côté compteur;

g) M_N ainsi que les limites L_N et U_N sont stockés avec une taille de données similaire à la valeur du STN complet de 32 bits et ne sont pas transmis dans le jeton.

Le Tableau 3 donne un exemple de définition de L_N et U_N pour chaque SubClass de jeton en considérant le nombre de jetons à émettre.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Tableau 3 – Exemple de définition de L_N et U_N pour chaque SubClass

SubClass	L_N	U_N	Commentaire
0	$M_N + 1 - 3R/8$ ou L_N si la valeur de ($M_N + 1 - 3R/8$) est inférieure à la valeur actuelle de L_N	$M_N + R/8$	La plage comprise entre les limites (incluses) est normalement de $R/2$. Ceci définit une fenêtre d'acceptation. La limite haute étant quelque peu supérieure au nombre de transactions le plus élevé accepté, plusieurs futures transactions peuvent être saisies hors séquence. L_N doit être stocké. U_N peut être calculé à partir de M_N.
1	$M_N + 1$	$M_N + R/2$	Seules les transactions plus tardives que la dernière acceptée sont acceptées, mais il y a une plage de tolérance pour omettre des transactions de la séquence. L_N et U_N peuvent être calculés à partir de M_N.
8 et 10	$M_N + 1 - 3R/8$ ou L_N si la valeur de ($M_N + 1 - 3R/8$) est inférieure à la valeur actuelle de L_N	$M_N + R/8$	La plage comprise entre les limites (incluses) est normalement de $R/2$. Ceci définit une fenêtre d'acceptation. La limite haute étant quelque peu supérieure au nombre de transactions le plus élevé accepté, plusieurs futures transactions peuvent être saisies hors séquence. L_N doit être stocké. U_N peut être calculé à partir de M_N. L'acceptation du jeton de la SubClass 10 en suivant strictement la séquence d'émission par le POS/HES est assurée par le paramètre appelé "FunctionIndex". Une description est donnée en 6.1.9.
9	$M_N + 1$	$M_N + R/2$	Seules les transactions plus tardives que la dernière acceptée sont acceptées, mais il y a une plage de tolérance pour omettre des transactions de la séquence. L_N et U_N peuvent être calculés à partir de M_N.
11	$M_N + 1 - 3R/8$ ou L_N si la valeur de ($M_N + 1 - 3R/8$) est inférieure à la valeur actuelle de L_N	$M_N + R/8$	La plage comprise entre les limites (incluses) est normalement de $R/2$. Ceci définit une fenêtre d'acceptation. La limite haute étant quelque peu supérieure au nombre de transactions le plus élevé accepté, plusieurs futures transactions peuvent être saisies hors séquence. L_N doit être stocké. U_N peut être calculé à partir de M_N.
X …			X est une future SubClass qui peut utiliser des méthodes supplémentaires pour appliquer une ou plusieurs règles différentes. De nouvelles sous-classes pourraient être définies avec le même comportement que celles existantes; ceci pourrait permettre à des sous-classes distinctes d'utiliser des règles d'acceptation identiques sans interaction entre leurs numéros, comme ce serait le cas si elles étaient affectées à la même sous-classe.
NOTE Pour les SubClass 8 et 10, seule 1 rangée est utilisée dans la colonne de dessus. Cette disposition est destinée à indiquer que ces deux sous-classes partagent une valeur STN commune, L_N et U_N.			

Dans l'exemple ci-dessus, R vaut 1 024 et la plage de TSTN va de 0 à 1 023; par conséquent, si STN vaut 1 040 alors TSTN = 1 040 MOD 1 024 = 16.

Pour la SubClass 0 de Class 5 et la SubClass 8 de Class 5, le processus de validation du STN et de déduction de MS(N) est indiqué dans le Tableau 4.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Tableau 4 – Processus de validation du STN et de déduction du MS(N)

Description du contrôle
Si LS(L_N) < LS(U_N) [de sorte que MS(L_N) = MS(U_N)] alors: si TSTN < LS(L_N) alors N est rejeté sinon si TSTN > LS(U_N) alors N est rejeté sinon N est accepté et MS(N) = MS(L_N) = MS(U_N); Si LS(L_N) > LS(U_N) [de sorte que MS(L_N) + 1 = MS(U_N)] alors: si TSTN ≥ LS(L_N) alors MS(N) = MS(L_N) sinon si TSTN ≤ LS(U_N) alors MS(N) = MS(U_N) sinon N est rejeté;

Exemple 1:

• STN de la SubClass 0 du dernier jeton accepté: 407 (décimal) = 0x00000197 (hexadécimal) (voir Tableau 5 et Tableau 6).

Tableau 5 – Exemple (a) de dernier jeton accepté

Valeur STN du dernier jeton de Subclass 0 accepté (décimale)	L_N (décimale)	U_N (décimale)	Valeur STN d'un nouveau jeton de SubClass 0 (décimale)	LS(L_N)	LS(U_N)	Valeur TSTN d'un nouveau jeton de SubClass 0 (décimale)	Vérification des conditions	MS(N)
407	24	535	408	24	535	408	LS(L_N)< LS(U_N) est VRAI.	MS(L_N) = MS(U_N) = 0

Tableau 6 – Exemple (b) de dernier jeton accepté

STN	0.....23	24.....535	536.....1 023
TSTN	0.....23	24.....535	536.....1 023
MS déduit avec état d'acceptation ou de rejet du jeton	Rejeté	Accepté, MS = 0	Rejeté

Exemple 2:

• STN de la SubClass 0 du dernier jeton accepté: 1023 (décimal) = 0x000003FF (hexadécimal) (voir Tableau 7 et Tableau 8).

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Tableau 7 – Exemple (c) de dernier jeton accepté

Valeur STN du dernier jeton de Subclass 0 accepté (décimale)	L_N (décimale)	U_N (décimale)	Valeur STN d'un nouveau jeton de SubClass 0 (décimale)	LS(L_N)	LS(U_N)	Valeur TSTN d'un nouveau jeton de SubClass 0 (décimale)	Vérification des conditions	MS(N)
1 023	640	1 151	1 024	640	127	0	LS(L_N)< LS(U_N) est FAUX. LS(L_N)> LS(U_N) est VRAI. TSTN ≤ LS(U_N) est VRAI.	MS(N) = MS(U_N) = 1

Tableau 8 – Exemple (d) de dernier jeton accepté

STN	0.....639	640.....1 023	1 024..... 151	1 152 et plus
TSTN	0.....639	640.....1 023	0.....127	128 et plus
MS déduit avec état d'acceptation ou de rejet du jeton	Rejeté	Accepté, MS = 0	Accepté, MS = 1	Rejeté

Si le jeton est accepté et si le STN de valeur N est supérieur au plus grand STN précédemment accepté par le compteur, les limites LN et UN sont avancées conformément au tableau des plages de sous-classes (Tableau 3).

6.1.9 FunctionIndex

FunctionIndex est un entier non signé de 32 bits qui est utilisé pour appliquer l'ordre d'acceptation de jetons particuliers ou de groupes de jetons. Chaque fois que le système TransferCredit génère un jeton dont le type doit être saisi dans le compteur (généralement un jeton destiné à changer le fonctionnement ou le tarif du compteur, par opposition à un simple transfert de crédit), il augmente FunctionIndex de 1. Le compteur exige que chaque jeton de SubClass 8 saisi ait un FunctionIndex inférieur ou égal au plus grand FunctionIndex accepté, et que chaque jeton de SubClass 10 ait un FunctionIndex supérieur de 1 à la plus grande valeur déjà stockée dans la mémoire du compteur. Par conséquent, en donnant respectivement à deux jetons Q et R de ce groupe fonctionnel FunctionIndex les valeurs q = p+1 et r = p+2, où p est le plus grand FunctionIndex précédemment accepté pour le jeton P, le compteur n'accepte pas le jeton tant que le jeton Q n'a pas été saisi et accepté par le compteur.

NOTE $q = p+1$ assure que Q peut être saisi lorsque le FunctionIndex du compteur est p, et Q a pour effet de le faire avancer à partir de p, alors que R ne peut pas être saisi avant Q lorsque FunctionIndex est toujours égal à p (car $r = p+2$ et non p+1), mais R peut être saisi après Q car $r = q+1$.

Pour ce qui concerne le jeton, FunctionIndex est une valeur partiellement implicite (plutôt que d'être explicitement et entièrement transférée à l'intérieur du jeton) qui est maintenue par le compteur, dans le cadre du processus d'application du compteur, dans un registre interne, ainsi que dans le HES et est utilisée pour générer et valider le MAC.

A chaque génération d'un SuperToken dont le type doit être saisi dans le compteur, FunctionIndex est augmenté de 1.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Dès que le compteur voit un jeton de Subclass TransferCredit+Function, il ajoute 1 au registre interne du FunctionIndex afin de valider le SuperToken.

Les sous-jetons contenus dans le SuperToken sont contrôlés par le STN dans le premier jeton pour éviter toute rediffusion. Une valeur de compteur (STN) est présumée pour le SuperToken complet.

L'ordre des sous-jetons dans le SuperToken est exclusivement géré par les 2 premiers bits des jetons suivants indiquant les jetons restants qui arriveront.

Les sous-jetons contenus dans un SuperToken doivent être saisis dans un ensemble contigu sans autres jetons ou sous-jetons dans la séquence.

Il est nécessaire que les sous-jetons, y compris le premier, soient saisis dans l'ordre correct. Comme le CheckDigit est calculé en incluant les sous-jetons précédents, il N'EST PAS permis de saisir les sous-jetons 2, 3 ou 4 dans un autre ordre.

Pour valider un tel jeton fonctionnel lorsque le jeton précédent accepté avant un FunctionIndex c, la valeur (c+1) est à utiliser.

Le premier jeton d'un type qui doit être saisi dans le compteur en respectant strictement la séquence doit avoir un nombre de groupes de FunctionIndex égal à zéro.

6.1.10 Lien entre le FunctionIndex et le STN

En partant du STN lui-même, le FunctionIndex est calculé côté compteur. A cet effet, la valeur maximale du STN d'un jeton SubClass, déjà accepté par le compteur, est à enregistrer dans un tableau dans un élément correspondant au nombre dérivé de "FunctionIndex" ou de "FunctionIndex and Maximum number of array elements in the array" à chaque incrémentation du FunctionIndex.

Le lien entre le FunctionIndex et le STN est établi pour obtenir l'acceptation de jeton de SubClass 8 côté compteur sans ignorer les jetons de SubClass 10. A chaque authentification réussie d'un jeton de SubClass 10, "STN value of the token – 1" est enregistré en mémoire dans un tableau correspondant au numéro d'élément équivalent à {"FunctionIndex" -1} ou à {"Function Index MOD number of array elements"-1}. Ce tableau est également appelé TokenGroupArray.

Par exemple, pour un compteur donné, un total de 7 jetons sont émis depuis l'installation du compteur et sur ces 7 jetons, les jetons ayant pour valeur STN 0x00000001,0x00000005 et 0x00000008 sont de SubClass 10, puis les 10 éléments du TokenGroupArray ont les valeurs suivantes:

0x00000000; 0x00000004; 0x00000007; 0x00000088; 0x00000000; 0x00000000; 0x00000000; 0x00000000; 0x00000000; 0x00000000.

Maintenant, avec les données ci-dessus, le FunctionIndex du compteur a stocké une valeur de 0x00000003.

Par conséquent, si un jeton de SubClass 8 manqué a une valeur TSTN de 0x00000006 et que le STN déduit est 0x00000006, alors en comparant le STN aux valeurs STN déjà stockées dans les éléments du tableau du TokenGroupArray, le "FunctionIndex" réel associé à ce jeton de SubClass peut être déterminé. Si le jeton manqué ayant un STN saisi au compteur de 0x00000006, ce compteur analyse le tableau et vérifie la condition associée à chaque élément et la condition deviendra VRAI pour l'élément du tableau portant le numéro 0x02. Par conséquent, le FunctionIndex du jeton arrivé est 0x00000002 et ceci est utilisé pour en déduire le TMAC du jeton.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Ainsi, sur la base des deux cas suivants:

- Cas 1: Le FunctionIndex actuel du compteur est inférieur ou égal au nombre d'éléments du TokenGroupArray;
- Cas 2: Le FunctionIndex actuel du compteur est supérieur au nombre d'éléments du TokenGroupArray.

En démarrant l'analyse à partir de l'élément du tableau de départ approprié jusqu'à ce que la valeur STN du jeton arrivé soit inférieure ou égale à la valeur stockée dans l'élément du tableau et en appliquant une formule appropriée, il est possible d'obtenir la valeur du FunctionIndex.

Pour le cas 1:

En commençant par l'élément d'élément 0, analyser et vérifier l'instant où la valeur STN du jeton arrivé est inférieure ou égale à la valeur dans l'élément du tableau. Si cette condition est vraie, alors la valeur du FunctionIndex est égale au numéro de l'élément du tableau.

Pour le cas 2:

En commençant par l'élément du tableau "N" (où N = la valeur actuelle du FunctionIndex MOD tous les éléments dans le tableau), analyser et vérifier l'instant où la valeur STN du jeton arrivé est inférieure ou égale à la valeur stockée dans l'élément du tableau. Si cette condition est vraie, alors la valeur du FunctionIndex est égale à la valeur actuelle du FunctionIndex plus le numéro de l'élément du tableau actuel moins le nombre d'éléments dans le tableau moins 1.

Telle est la façon dont le STN fournit lui-même le détail du FunctionIndex auquel il appartient.

Par exemple: si le STN maximal d'un jeton déjà accepté dans le groupe fonctionnel 1(SubClass 0 pour les jetons non chiffrés ou SubClass 8 "TransferCredit" et SubClass 10 "TransferCredit + Function" pour les jetons chiffrés) est 100d, alors le nouveau STN du type de jeton doit être plus élevé que 100d s'il s'avère nécessaire de changer le FunctionIndex.

Pour le STN maximal, la valeur du type de jeton de SubClass 8 "TransferCredit" déjà accepté avec le FunctionIndex 0d est 100d.

A la première émission du jeton de SubClass 10 "TransferCredit + Function", la plage du jeton à accepter dans le futur est calculée comme suit:

a) Plage complète pour le jeton:

   $L_N = M_N + 1 - 3R/8$,

   $U_N = M_N + R/8$

   M_N est la valeur maximale de N du jeton déjà accepté.

   L_N est la limite basse du STN N qui est valide.

   U_N est la limite haute du STN N qui est valide.

   C'est-à-dire que la valeur STN du dernier jeton accepté est 100d, L_N = 0d, U_N = 228d et M_N = 100d.

b) Si le jeton de SubClass 10 "Transfer Credit + Function" est généré avec le FunctionIndex 1d et la valeur STN (c'est-à-dire que N = 101d) et lorsque le MAC et tous les autres critères de validation du jeton sont déterminés comme étant valides après le processus de validation des jetons, (M_N – 1)d (c'est-à-dire 100d) est stocké en mémoire à l'élément 0 du TokenGroupArray (qui conserve l'enregistrement du M_N correspondant au FunctionIndex, chaque élément du tableau de 4 octets) et U_N (c'est-à-dire 228d) est stocké au 1er élément de l'UN à 4 octets en maintenant le tableau en mémoire. Par conséquent, les jetons de SubClass 8 "TransferCredit" du type ayant une valeur STN de LN à MN (c'est-à-dire de 0d

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

à 100d) appartiennent au FunctionIndex 0d et les types de jetons "TransferCredit" ayant des valeurs STN appartenant au FunctionIndex 1d sont MN+1 à UN (c'est-à-dire 101d à 228d). Par ailleurs, dans le cas ci-dessus, si UN change en raison de l'acceptation du ou des jetons "TransferCredit" sans modification de valeur du FunctionIndex, alors le compteur met également à jour le 1er élément du tableau avec la nouvelle valeur UN.

A l'étape suivante, le système de vente doit générer un autre jeton TransferCredit avec le FunctionIndex 1d uniquement et une valeur STN dans la plage de 102d à 228d (la valeur STN présumée du jeton TranferCredit est 101 avec un FunctionIndex incrémenté de la valeur 1).

Il convient de mémoriser le nombre d'éléments dans le TokenGroupArray, qui conserve l'enregistrement "relating FunctionIndex and STN", en tenant compte des trois facteurs suivants:

- Quelle est la fréquence des changements de tarif sur le marché particulier de la distribution?
- Quelle est la durée de vie attendue du compteur?
- Combien de jetons de Class 5 et de SubClass 10 "TranferCredit + Function" sont susceptibles d'être émis à la suite des changements de tarif?

6.1.11 SingleTokenPayload

SingleTokenPayload est une valeur de 61 bits composée des éléments suivants:

a) le TMAC de 32 bits qui est tronqué depuis la partie LS du MAC de 128 bits et placé dans les 32 bits LS des données du jeton de 61 bits (voir 6.2);

b) la valeur de 29 bits restante est remplie de 3 bits ayant la valeur 000b dans la position MS pour créer un nombre de 32 bits.

Le contenu et la signification des 29 bits MS de la valeur à 61 bits de la SingleTokenPayload varient en fonction de la SubClass du jeton (voir 6.2).

Par exemple: La SubClass 0 de Class 5 (voir 6.2.4.1), la SubClass 8 de Class 5 (voir 6.2.5.1) et la SubClass 9 de Class 5 (voir 6.2.5.2) ont une SingleTokenPayload avec un seul bloc de données.

Se reporter à la Figure 9 et au 6.1.15 pour obtenir des informations sur la dérivation du TMAC pour la SingleTokenPayload.

6.1.12 SuperTokenPayload

La SuperTokenPayload doit être une longueur variable de bits de l'APDU, qui dépend du nombre de blocs de données qui sont collectivement considérés pour obtenir un jeton.

La charge utile du SuperTokenPayload varie en fonction du nombre de blocs de données dans le SuperToken, c'est-à-dire 122 bits (61 bits + 61 bits), 183 bits (61 bits + 61 bits + 61 bits), 244 bits (61 bits + 61 bits + 61 bits + 61 bits) pour 2, 3 ou 4 blocs de données respectivement. Bien que la taille de la SuperTokenPayload soit indiquée en bits ci-dessus pour un nombre discret de blocs de données, les 61 bits associés à chaque bloc de données sont stockés dans un espace de stockage de 8 octets, en conservant chaque bloc de 3 bits MS sous la forme 000b.

Par exemple: SubClass 10 de Class 5 ayant la SuperTokenPayload (voir 6.2.5.3).

Se reporter à la Figure 10 et au 6.1.16 pour obtenir des informations sur la dérivation du TMAC pour la SuperTokenPayload.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.1.13 MessageAuthenticationCode (MAC) et TruncatedMAC (TMAC)

6.1.13.1 Calcul du MAC

Le calcul du MAC de 128 bits est effectué sur les éléments de données suivants pour la SingleTokenPayload, conformément à la Figure 5.

A la Figure 5, le bloc générique 0 est représenté sous la forme de 64 bits complets, mais seuls les 32 bits MS sont utilisés pour la construction de la charge utile de l'AAD selon la SubClass du jeton. Une description est également donnée en 6.1.14.

Figure 5 – Eléments de données génériques pour la construction de la charge utile de l'AAD pour la SingleTokenPayload

Le calcul du MAC de 128 bits est effectué sur les éléments de données suivants pour la SuperTokenPayload, conformément à la Figure 6.

A la Figure 6, le bloc générique 0 est représenté en totalité, mais seuls les 32 bits MS sont utilisés pour la construction de la charge utile de l'AAD selon la SubClass du jeton. Une description est également donnée en 6.1.14.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

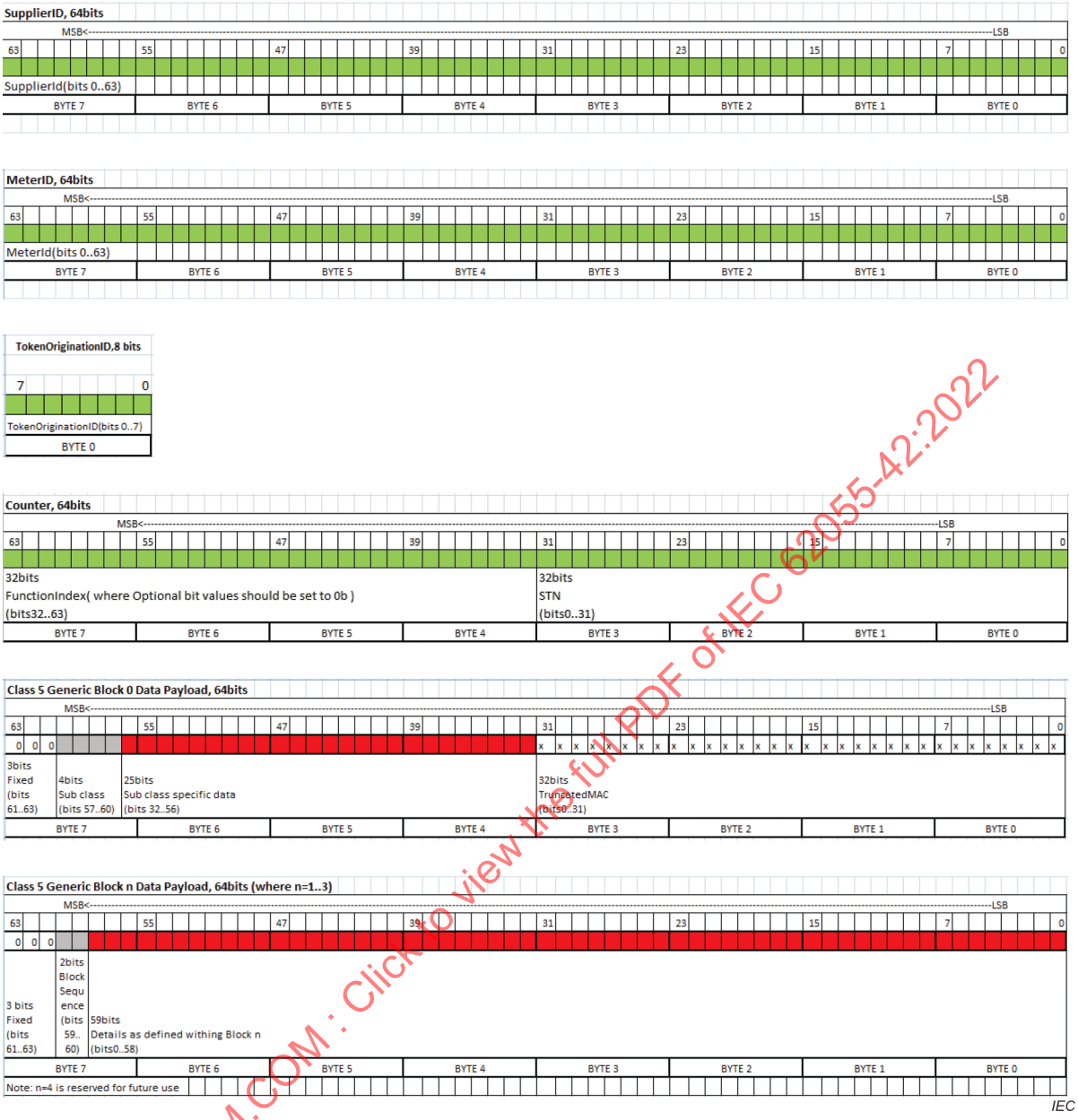


Figure 6 – Eléments de données génériques pour la construction de la charge utile de l'AAD pour la SuperTokenPayload

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

La Figure 7 fournit des informations sur la préparation de la charge utile de l'IV en utilisant les éléments de données InvocationField et SuppliedID.

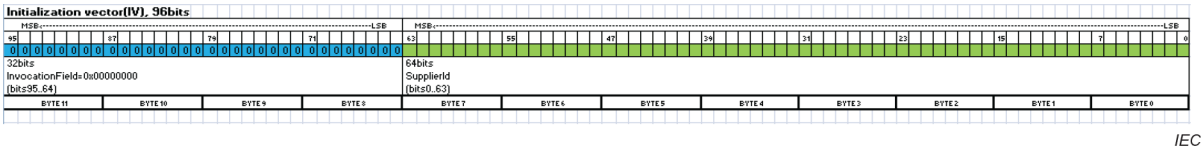


Figure 7 – Construction du InitializationVector (IV)

6.1.13.2 Construction du GMAC

La Figure 8 représente les éléments de données d'entrée du moteur GMAC pour déterminer la valeur MAC. Les 32 bits LS de cette valeur MAC sont utilisés comme TMAC dans la construction de l'APDU.

L'algorithme du GMAC, reposant sur l'utilisation de l'AES-128, est utilisé pour calculer le MAC selon les spécifications de la NIST SP 800-38D: 2007.

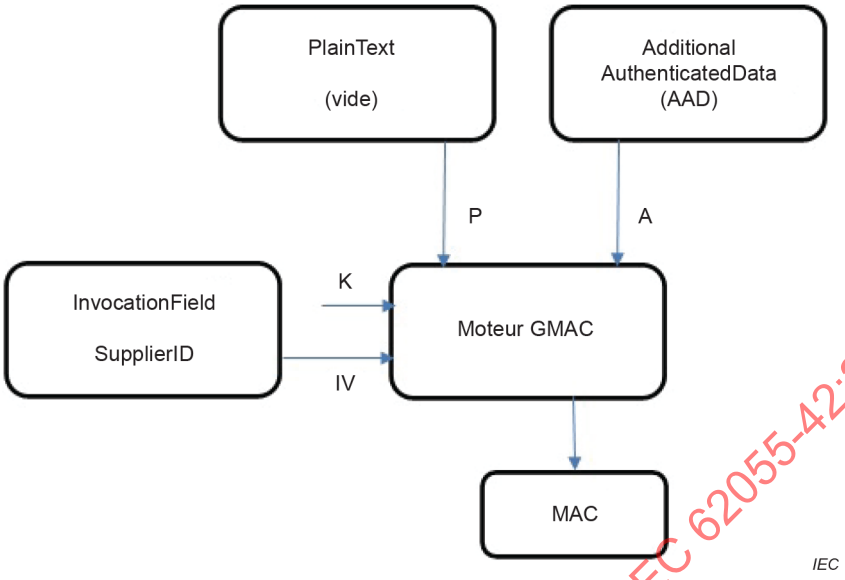


Figure 8 – Construction du GMAC

6.1.14 AdditionalAuthenticationData (AAD)

L'AAD doit être obtenu par la concaténation suivante:

AAD = MessageIdentifier || 32 bits MS de la charge utile des données du bloc générique 0 || 64 bits de la charge utile (facultative) du bloc générique 1 || 64 bits de la charge utile (facultative) du bloc générique 2 || 64 bits de la charge utile (facultative) du bloc générique 3.

NOTE 1 Le nombre de charges utiles des blocs dans l'AAD dépend de la sous-classe et des données à envoyer dans la sous-classe.

NOTE 2 Les détails de la sous-classe et les numéros de séquence des blocs ne sont pas inclus dans l'AAD.

6.1.15 Préparation de l'AAD de la SingleTokenPayload, dérivation du TMAC et préparation de l'APDU

Le présent paragraphe décrit la préparation de l'AAD pour la SingleTokenPayload basée sur les éléments de données d'un jeton de SubClass 8, la dérivation du TMAC et donc la mise à jour de l'APDU complète pour le jeton de SubClass 8.

La Figure 9 représente un exemple de dérivation du TMAC pour un jeton de SubClass 8, en considérant l'approche représentée à la Figure 8.

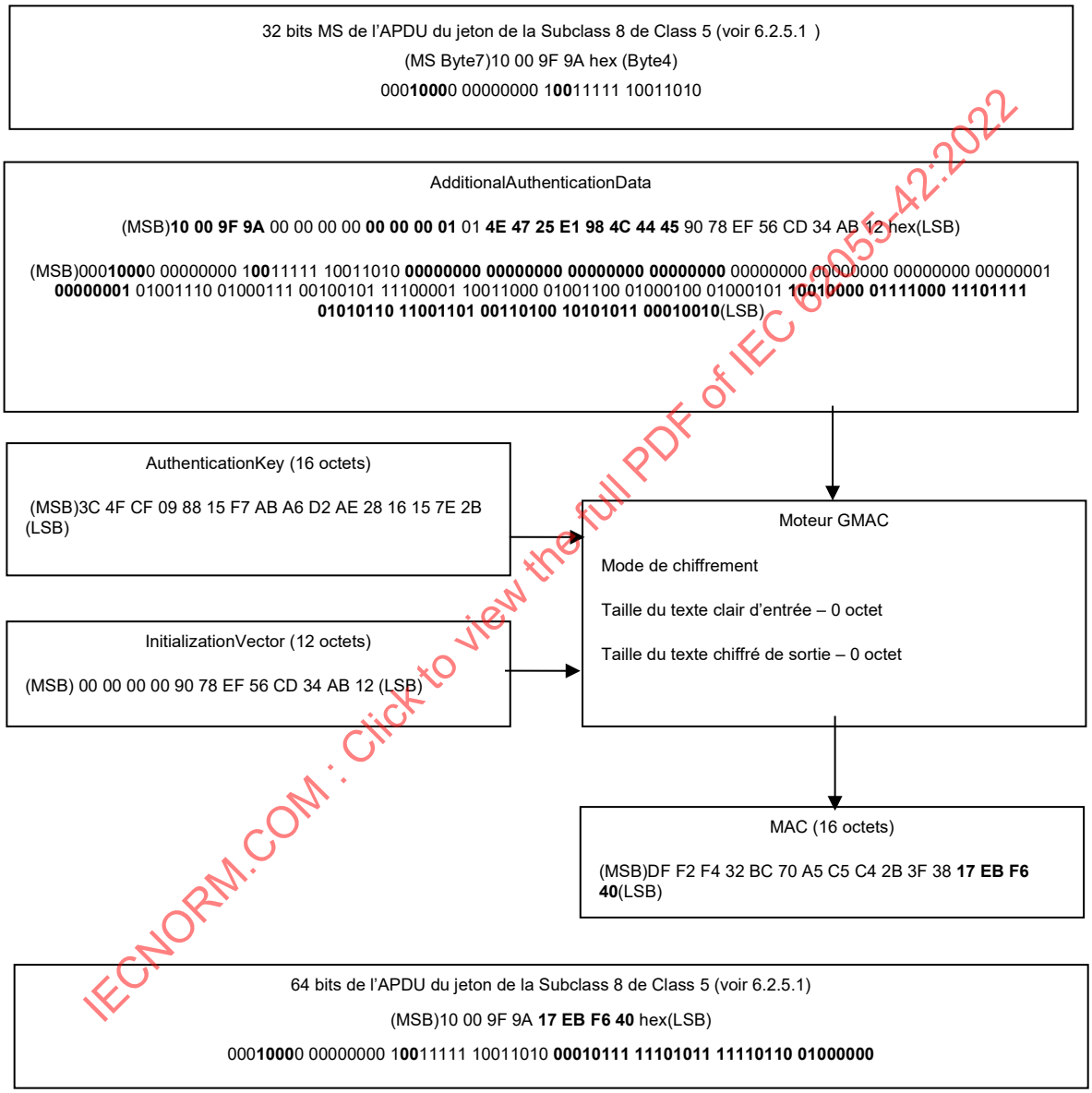
Les éléments de données présentés ci-dessous se présentent sous forme de valeur de base ou de valeur hexadécimale:

- SupplierID: 90 78 EF 56 CD 34 AB 12;
- MeterID: 45 44 4C 98 E1 25 47 4E;
- TokenOriginationID: 0x01;
- STN: 00 00 00 01;
- FunctionIndex: 00 00 00 00;
- Quantité à transférer: 8090d.

Les instances dérivées d'IV et de MessageIdentifier sont comme suit:

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

- Tampon InitializationVector: 12 AB 34 CD 56 EF 78 90 00 00 00 00;
- Tampon MessageIdentifier: **12 AB 34 CD 56 EF 78 90** 45 44 4C 98 E1 25 47 4E **01** 01 00 00 00 **00 00 00 00**;
- 32 bits MS de la SingleTokenPayload: Block1(Byte7)10 00 9F 9A (Byte4);
- Tampon AdditionalAuthenticationData: **12 AB 34 CD 56 EF 78 90** 45 44 4C 98 E1 25 47 4E **01** 01 00 00 00 **00 00 00 00** 9A 9F 00 10;
- AuthenticationKey: 3C 4F CF 09 88 15 F7 AB A6 D2 AE 28 16 15 7E 2B.



SHAPE * MERGEFORMAT

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Figure 9 – Exemple de dérivation du TMAC de la SubClass 8 de Class 5 et de préparation d'APDU complète

6.1.16 Préparation de l'AAD de la SuperTokenPayload, dérivation du TMAC et préparation de l'APDU

Le présent paragraphe décrit la préparation de l'AAD pour la SuperTokenPayload basée sur les éléments de données d'un jeton de SubClass 10, la dérivation du TMAC et donc la mise à jour de l'APDU complète pour le jeton de SubClass 10.

La Figure 10 représente un exemple de dérivation du TMAC pour un jeton de SubClass 10, en considérant l'approche représentée à la Figure 8.

Les éléments de données présentés ci-dessous se présentent sous forme de valeur de base ou de valeur hexadécimale (stockée en mémoire au format petit-boutiste):

- SupplierID: 90 78 EF 56 CD 34 AB 12;
- MeterID: 45 44 4C 98 E1 25 47 4E;
- TokenOriginationID: 0x01;
- STN: 00 00 00 02;
- FunctionIndex: 00 00 00 01;
- Quantité à transférer: 8090d;
- Tarif à 8 tranches: 355, 600, 900, 1200, 1600, 2000, 2600;
- ActivationMonth: Février 2019.

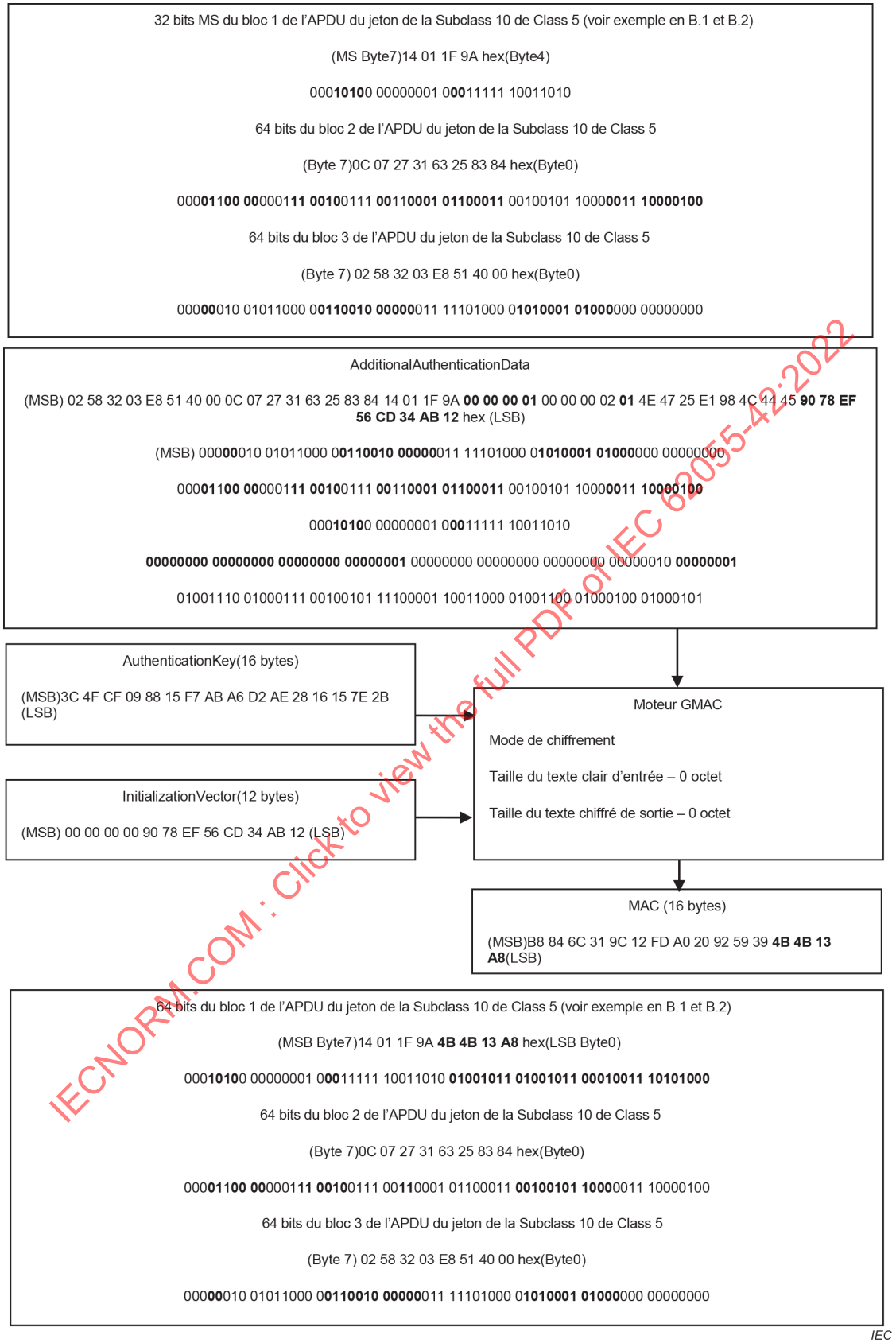
Les instances dérivées d'IV et de MessageIdentifier sont comme suit:

- Tampon InitializationVector: 12 AB 34 CD 56 EF 78 90 00 00 00 00;
- Tampon MessageIdentifier: **12 AB 34 CD 56 EF 78 90** 45 44 4C 98 E1 25 47 4E **01** 02 00 00 00 **01 00 00 00**;
- SuperTokenPayload:
  - Block 1 (Byte7)14 01 1F 9A(Byte4);
  - Block 2 (Byte7)0C 07 27 31 63 25 83 84(Byte0);
  - Block 3 (Byte7)02 58 32 03 E8 51 40 00(Byte0).

Tampon AdditionalAuthenticationData:

- **12 AB 34 CD 56 EF 78 90** 45 44 4C 98 E1 25 47 4E **01** 02 00 00 00 **01 00 00 00** 9A 1F 01 14 84 83 25 63 31 27 07 0C 00 40 51 E8 03 32 58 02;
- AuthenticationKey: 3C 4F CF 09 88 15 F7 AB A6 D2 AE 28 16 15 7E 2B.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022



IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Figure 10 – Exemple de dérivation du TMAC de la SubClass 10 de Class 5 et de préparation d'APDU complète

6.1.17 Décalage

Dans le calcul des jetons de Class 4 et ci-dessus, il est nécessaire d'appliquer différents décalages afin d'accéder au domaine de jeton correct lors de l'assemblage de l'APDU et de la transformation en TCDU. Le Tableau 9 définit ces décalages et leurs valeurs associées aux valeurs minimales et maximales des différentes classes de jetons.

Les valeurs indiquées sont aux formats décimal et hexadécimal. Les valeurs décimales peuvent être utilisées pour déterminer, par examen, la classe à laquelle appartient un jeton décimal, mais il peut s'avérer pratique d'ajouter ou de soustraire un décalage par arithmétique binaire et le présent document n'impose pas d'utiliser l'arithmétique binaire ou décimale.

Tableau 9 – Constantes numériques et leur finalité

Constante	Valeur décimale	Valeur hexadécimale	Finalité
-	00 000 000 000 000 000 000	0x0 0000 0000 0000 0000	valeur MIN d'un jeton de Class 0, 1, 2, 3
K_Class_0_1_ 2_3_MAX	73 786 976 294 838 206 463	0x3 FFFF FFFF FFFF FFFF	valeur MAX d'un jeton de Class 0, 1, 2, 3
K_Class _4_MIN	73 786 976 294 838 206 470	0x4 0000 0000 0000 0006	valeur MIN d'un jeton de Class 4
K_Class_4_MAX	73 941 569 907 863 060 479	0x4 0225 3A12 6CE3 FFFF	valeur MAX d'un jeton de Class 4
K_Class 5_OFFSET (19 chiffres)	7 394 156 990 786 306 048	0x669D 529B 714A 0000	Constante à ajouter au jeton de Class 5
K_Class_5_MIN	73 941 569 907 863 060 480	0x4 0225 3A12 6CE4 0000	valeur MIN d'un jeton de Class 5
K_Class_5_MAX	96 999 999 999 999 999 999	0x5 4225 3A12 6CE3 FFFF	valeur MAX d'un jeton de Class 5
K_Class_6_MIN	97 000 000 000 000 000 000	0x5 4225 3A12 6CE4 0000	Les classes supplémentaires restant à spécifier débutent ici
K_Class_X_MAX	99 999 999 999 999 999 999	0x5 6BC7 5E2D 630F FFFF	Fin du domaine des jetons (plus grand nombre décimal possible à 20 chiffres)

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.2 Jetons

6.2.1 Définition et format des jetons

La charge utile des jetons comporte la valeur de 29 bits plus le TMAC de 32 bits dans l'APDU (voir 6.1.1, 6.1.11 et 6.1.12) pour former une SingleTokenPayload binaire de 61 bits contenant un certain nombre de champs d'éléments de données plus petits, spécifiés dans le présent paragraphe. Les domaines de jetons spécifiés dans le présent document ne doivent pas chevaucher le cadre défini par l'IEC 62055-41:2018 existante, relative à la STS, et doivent réellement établir un nouveau domaine d'action en dehors de celui des classes 0, 1, 2 et 3 de jetons STS. Un certain nombre de nouveautés par rapport à l'IEC 62055-41:2018 actuellement spécifiée sont exigées pour le jeton de Class 5, lesquelles sont principalement la possibilité d'utiliser un jeton non chiffré, un jeton chiffré et un TMAC intégré pour l'authentification des jetons.

La définition et le format des informations pour les jetons sont indiqués dans le Tableau 10.

Tableau 10 – Définition et format des jetons

Nom de l'élément de données	par exemple Amount, MAC, Class, SubClass, etc.
Format des données	par exemple binaire, décimal, etc.
Nombre de bits	par exemple 15 bits, etc.
Plage de valeurs	par exemple 0-9, 1,3, etc.

Les jetons sont définis dans des classes dépendant de leur fonction, de la façon suivante:

- Class 0 – exclusivement définie dans l'IEC 62055-41:2018;
- Class 1 – exclusivement définie dans l'IEC 62055-41:2018;
- Class 2 – exclusivement définie dans l'IEC 62055-41:2018;
- Class 3 – exclusivement définie dans l'IEC 62055-41:2018;
- Class 4 – exclusivement définie dans le présent document;
- Class 5 – exclusivement définie dans le présent document.

6.2.2 Class 4: RESERVEE POUR UNE FUTURE AFFECTATION

Cette classe de plages de jetons est réservée pour une future allocation par le WG15 de l'IEC/TC13 et ne doit pas être utilisée à une autre fin.

6.2.3 Jetons de Class 5

6.2.3.1 Généralités

La plage de jetons de la Class 5 est divisée en 16 types de jeton identifiés en utilisant le champ SubClass. Ces 16 types sont scindés en 2 catégories: les jetons non chiffrés et les jetons chiffrés. Les valeurs de SubClass de 0 à 7 doivent être utilisées pour les jetons non chiffrés et les valeurs de SubClass de 8 à 15 doivent être utilisées pour les jetons chiffrés comme indiqué dans le Tableau 11.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Tableau 11 – Affectation des SubClass de la Class 5

Jeton SubClass	Class 5 de jeton	
	NON CHIFFRE	CHIFFRE
0	Jeton TransferCredit	Non admis
1	Réservé pour SpecialFunction	Non admis
2	Réservé	Non admis
3	Réservé	Non admis
4	Réservé	Non admis
5	Réservé	Non admis
6	Réservé	Non admis
7	Réservé	Non admis
8	Non admis	Jeton TransferCredit
9	Non admis	Jeton SpecialFunction
10	Non admis	Jeton ExtendedTransferCredit
11	Non admis	Jeton MeterGenerated
12	Non admis	Réservé
13	Non admis	Réservé
14	Non admis	Réservé
15	Non admis	Réservé

Le Tableau 12 indique les plages du domaine décimal de la charge utile de l'APDU pour le domaine binaire à 61 bits, telles qu'affectées pour les sous-classes au sein de la Class 5 (avant chiffrement), et le Tableau 13 indique l'APDU de jeton convertie dans le domaine décimal avec le décalage de Class 5 ajouté à l'équivalent décimal de l'APDU à 61 bits (avant que le chiffre de vérification soit ajouté pour former la TCDU).

Ainsi, un dispositif peut déterminer la SubClass par examen de la valeur décimale sans déchiffrement, les 4 bits MS (bits 57 à 60) du bloc de jeton de 61 bits, contenant la SubClass, ne devant pas être chiffrés. Si le jeton se compose de plusieurs blocs, alors le numéro de séquence dans les 2 bits MS (bits 59 et 60) des blocs suivants (pour les blocs autres que le 1er bloc) ne doit pas être chiffré.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Tableau 12 – Limites de SubClass pour l'APDU de Class 5 avant chiffrement

SubClass (binaire)	Valeur de charge utile minimale (décimale)	Valeur de charge utile minimale (hexadécimale)	Valeur de charge utile maximale (décimale)	Valeur de charge utile maximale (hexadécimale)
0000	000000000000000000	0000000000000000	144115188075855871	01FFFFFFFFFFFF
0001	144115188075855872	0200000000000000	288230376151711743	03FFFFFFFFFFFF
0010	288230376151711744	0400000000000000	432345564227567615	05FFFFFFFFFFFF
0011	432345564227567616	0600000000000000	576460752303423487	07FFFFFFFFFFFF
0100	576460752303423488	0800000000000000	720575940379279359	09FFFFFFFFFFFF
0101	720575940379279360	0A00000000000000	864691128455135231	0BFFFFFFFFFFFF
0110	864691128455135232	0C00000000000000	1008806316530991103	0DFFFFFFFFFFFF
0111	1008806316530991104	0E00000000000000	1152921504606846975	0FFFFFFFFFFFFF
1000	1152921504606846976	1000000000000000	1297036692682702847	11FFFFFFFFFFFF
1001	1297036692682702848	1200000000000000	1441151880758558719	13FFFFFFFFFFFF
1010	1441151880758558720	1400000000000000	1585267068834414591	15FFFFFFFFFFFF
1011	1585267068834414592	1600000000000000	1729382256910270463	17FFFFFFFFFFFF
1100	1729382256910270464	1800000000000000	1873497444986126335	19FFFFFFFFFFFF
1101	1873497444986126336	1A00000000000000	2017612633061982207	1BFFFFFFFFFFFF
1110	2017612633061982208	1C00000000000000	2161727821137838079	1DFFFFFFFFFFFF
1111	2161727821137838080	1E00000000000000	2305843009213693951	1FFFFFFFFFFFFF

Tableau 13 – Limites de SubClass pour les jetons de Class 5, TCDU après chiffrement (si applicable) et ajout du décalage (sans CheckDigit)

SubClass – binaire	Valeur minimale – décimale	Valeur minimale – hexadécimale	Valeur maximale – décimale	Valeur maximale – hexadécimale
0000	7394156990786306048	669D529B714A0000	7538272178862161919	689D529B7149FFFF
0001	7538272178862161920	689D529B714A0000	7682387366938017791	6A9D529B7149FFFF
0010	7682387366938017792	6A9D529B714A0000	7826502555013873663	6C9D529B7149FFFF
0011	7826502555013873664	6C9D529B714A0000	7970617743089729535	6E9D529B7149FFFF
0100	7970617743089729536	6E9D529B714A0000	8114732931165585407	709D529B7149FFFF
0101	8114732931165585408	709D529B714A0000	8258848119241441279	729D529B7149FFFF
0110	8258848119241441280	729D529B714A0000	8402963307317297151	749D529B7149FFFF
0111	8402963307317297152	749D529B714A0000	8547078495393153023	769D529B7149FFFF
1000	8547078495393153024	769D529B714A0000	8691193683469008895	789D529B7149FFFF
1001	8691193683469008896	789D529B714A0000	8835308871544864767	7A9D529B7149FFFF
1010 [1)]	8835308871544864768	7A9D529B714A0000	8979424059620720639	7C9D529B7149FFFF
1011 [1)]	8979424059620720640	7C9D529B714A0000	9123539247696576511	7E9D529B7149FFFF
1100	9123539247696576512	7E9D529B714A0000	9267654435772432383	809D529B7149FFFF
1101	9267654435772432384	809D529B714A0000	9411769623848288255	829D529B7149FFFF
1110	9411769623848288256	829D529B714A0000	9555884811924144127	849D529B7149FFFF
1111	9555884811924144128	849D529B714A0000	9699999999999999999	869D529B7149FFFF

NOTE 1 La SubClass 10 et la SubClass 11 ayant plus de 1 bloc de données d'APDU et donc plus de 20 chiffres de TCDU, la plage du tableau ci-dessus est uniquement applicable pour la TCDU du bloc de données 1.

NOTE 2 Les valeurs indiquées sont valables avant l'ajout du CheckDigit.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Le Tableau 14 indique les limites pour les TCDU dans toutes les 16 sous-classes au sein de la Class 5. Etant donné la nature du chiffre de vérification, il est probable que les représentations décimales des jetons minimal et maximal réels ne se terminent pas par 0 et 9 respectivement, car pour chaque charge utile de jeton de 19 chiffres, il n'y a qu'un seul chiffre de vérification possible, et donc un seul sur dix TCDU de 20 chiffres est légitime. Cependant, l'espace décimal

complet dans le domaine à 20 chiffres est quand-même réservé pour les sous-classes appropriées. De plus, il NE DOIT PAS être permis d'utiliser des valeurs de jeton dans les limites qui n'ont pas de chiffre de vérification valide pour une quelconque autre finalité, y compris à des fins spécifiques au fabricant.

Tableau 14 – Limites de SubClass de la Class 5 pour la TCDU (espace réservé)

SubClass – binaire	TCDU actuelle minimale	TCDU actuelle maximale
0000	73941569907863060480	75382721788621619199
0001	75382721788621619200	76823873669380177919
0010	76823873669380177920	78265025550138736639
0011	78265025550138736640	79706177430897295359
0100	79706177430897295360	81147329311655854079
0101	81147329311655854080	82588481192414412799
0110	82588481192414412800	84029633073172971519
0111	84029633073172971520	85470784953931530239
1000	85470784953931530240	86911936834690088959
1001	86911936834690088960	88353088715448647679
1010 [1)]	88353088715448647680	89794240596207206399
1011 [1)]	89794240596207206400	91235392476965765119
1100	91235392476965765120	92676544357724323839
1101	92676544357724323840	94117696238482882559
1110	94117696238482882560	95558848119241441279
1111	95558848119241441280	96999999999999999999
NOTE La SubClass 10 et la SubClass 11 ayant plus de 1 bloc de données d'APDU et donc plus de 20 chiffres de TCDU (TCDU à 40 ou 60 ou 80 chiffres), la plage est uniquement applicable pour les 20 premiers chiffres de la TCDU.		

6.2.3.2 FunctionalClass liée à la SubClass de jeton de Class 5 et cas d'utilisation associé

Chaque acceptation de jeton de SubClass côté compteur a un certain comportement ou une certaine approche fonctionnelle basée sur son cas d'utilisation pratique, comme l'indique le Tableau 15. Dans le présent document, ce comportement ou cette approche fonctionnelle est décrit par le numéro de FunctionalClass.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Tableau 15 – FunctionalClass liée à la SubClass et cas d'utilisation associés

SubClass	FunctionalClass	Comportement	Cas d'utilisation
0	1	Il n'est pas nécessaire de saisir les jetons dans le compteur dans la séquence exacte dans laquelle ils ont été émis, ou il y a tout du moins une marge de manœuvre considérable (mais pas une liberté complète) dans l'ordre de saisie.	Si deux jetons TransferCredit sont émis vers un consommateur, ils peuvent alors être saisis dans le compteur dans un ordre quelconque.
8	1	Il n'est pas nécessaire de saisir les jetons dans le compteur dans la séquence exacte dans laquelle ils ont été émis, ou il y a tout du moins une marge de manœuvre considérable (mais pas une liberté complète) dans l'ordre de saisie.	Si deux jetons TransferCredit sont émis vers un consommateur, ils peuvent alors être saisis dans le compteur dans un ordre quelconque.
9	3	Les jetons doivent être saisis dans le compteur en respectant la séquence dans laquelle ils ont été émis, mais certains peuvent être omis sans affecter les transactions ultérieures. Ces transactions omises ne peuvent ensuite pas être saisies. Un STN séparé est maintenu pour les jetons SpecialFunction comparés aux jetons "TransferCredit" et "TransferCredit + Function", bien qu'ils utilisent le même format. De la même façon, un jeton SpecialFunction a son propre TSTN comparé à celui des jetons "TransferCredit" et "TransferCredit + Function". Tous les jetons SpecialFunction sont à saisir dans le compteur dans l'ordre si l'on souhaite que tous les jetons soient acceptés et qu'ils deviennent invalides une fois arrivés à expiration. Par conséquent, les jetons créés dans le futur peuvent être saisis et sont acceptés; en revanche, les jetons inutilisés plus anciens ne sont pas acceptés par le compteur.	Un jeton SpecialFunctionToken est remplacé par un autre SpecialFunctionToken émis par la suite. Le jeton le plus récent n'est pas accepté après le plus ancien.
10	1	Les jetons doivent être saisis dans le compteur en respectant strictement la séquence dans laquelle ils ont été émis sans omission.	Le consommateur a émis un jeton "TransferCredit + Function". Celui-ci doit être saisi avant tout jeton "TransferCredit"/ "TransferCredit + Function" acheté par la suite. Si la règle qui précède est respectée, il est alors accepté.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Les jetons de la SubClass 8 et de la SubClass 10 ont tous deux une information de quantité et il convient donc de maintenir un STN commun pour les deux types de sous-classes et de maintenir les paramètres L_N, U_N et M_N en commun pour ces deux types de jetons de SubClass. Ainsi, STN, L_n, U_n et M_n sont à maintenir au niveau de la SubClass fonctionnelle côté compteur et il convient d'établir une mise en correspondance entre la SubClass et le numéro de FunctionalClass côté compteur.

6.2.4 Classe 5: jetons non chiffrés

6.2.4.1 SubClass 0: jeton TransferCredit

Le Tableau 16 indique la structure d'un jeton TransferCredit de SubClass 0.

Tableau 16 – SubClass 0: jeton TransferCredit

SubClass	TSTN	AMTConfig	AMT	TMAC
Binaire	Binaire	Binaire	Binaire	Binaire
4 bits	10 bits	2 bits	13 bits	32 bits
0 = TransferCredit non chiffré Voir 6.2.3	Numéro à 32 chiffres tronqué	0d – 3d Voir 6.3	0d – 8191d Voir 6.3	GMAC à 128 bits tronqué

AMTConfig détermine l'interprétation de l'AMT comme suit:

- 0d – Signifie que l'AMT va de 0d à 8191d par pas de 1;
- 1d – Signifie que l'AMT va de 0d à 819100d par pas de 100;
- 2d – Signifie que l'AMT va de 0d à 81910000d par pas de 10 000;
- 3d – Signifie que l'AMT va de 0d à 8191000000d par pas de 1 000 000.

6.2.4.2 SubClass 1: jetons SpecialFunction

Le présent paragraphe spécifie le format des jetons SpecialFunction qui sont affectés à la SubClass 1 au sein de la Class 5.

La structure de codage est à définir par le WG15 de l'IEC/TC13 et ne peut pas être utilisée à une autre fin.

6.2.4.3 SubClass 2: RESERVEE

Cette SubClass est réservée pour une future affectation par le WG15 de l'IEC/TC13 et ne peut pas être utilisée à une autre fin.

6.2.4.4 SubClass 3: RESERVEE

Cette SubClass est réservée pour une future affectation par le WG15 de l'IEC/TC13 et ne peut pas être utilisée à une autre fin.

6.2.4.5 SubClass 4: RESERVEE

Cette SubClass est réservée pour une future affectation par le WG15 de l'IEC/TC13 et ne peut pas être utilisée à une autre fin.

6.2.4.6 SubClass 5: RESERVEE

Cette SubClass est réservée pour une future affectation par le WG15 de l'IEC/TC13 et ne peut pas être utilisée à une autre fin.

6.2.4.7 SubClass 6: RESERVEE

Cette SubClass est réservée pour une future affectation par le WG15 de l'IEC/TC13 et ne peut pas être utilisée à une autre fin.

6.2.4.8 SubClass 7: RESERVEE

Cette SubClass est réservée pour une future affectation par le WG15 de l'IEC/TC13 et ne peut pas être utilisée à une autre fin.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.2.5 Classe 5: jetons chiffrés

6.2.5.1 SubClass 8: jeton TransferCredit

Le Tableau 17 indique la structure d'un jeton TransferCredit de SubClass 8.

Tableau 17 – SubClass 8: jeton TransferCredit

SubClass	TSTN	AMTConfig	AMT	TMAC
Binaire	Binaire	Binaire	Binaire	Binaire
4 bits	10 bits	2 bits	13 bits	32 bits
8 = 1000b Jeton TransferCredit chiffré Voir 6.2.3	Numéro à 32 chiffres tronqué	0d – 3d	0d – 8191d	GMAC à 128 bits tronqué

AMTConfig détermine l'interprétation de l'AMT comme suit:

- 0d – Signifie que l'AMT va de 0d à 8191d par pas de 1;
- 1d – Signifie que l'AMT va de 0d à 819100d par pas de 100;
- 2d – Signifie que l'AMT va de 0d à 81910000d par pas de 10 000;
- 3d – Signifie que l'AMT va de 0d à 8191000000d par pas de 1 000 000.

6.2.5.2 SubClass 9: jetons SpecialFunction

6.2.5.2.1 Généralités

Le présent paragraphe spécifie le format des jetons SpecialFunction qui sont affectés à la SubClass 9 au sein de la Class 5.

La structure de codage pour un jeton SpecialFunction est indiquée dans le Tableau 18.

Les champs de données SubClass, TSTN et TMAC dans ce jeton doivent être conformes aux définitions de 6.2.3, 6.1.7 et 6.1.13.

Tableau 18 – Class 5, SubClass 9: Jeton SpecialFunction

SubClass	TSTN	Données		TMAC
		ServiceType	RES	
4 bits	10 bits	5 bits	10 bits	32 bits
9 = 1001b – Ingénierie Voir 6.2.3	Numéro à 32 chiffres tronqué	Voir Tableau 19.	Réservé	MAC à 128 bits tronqué
NOTE 1 Le chiffre de vérification n'apparaît pas dans ce tableau car il s'applique à la représentation décimale du jeton lors de sa conversion en TCDU. NOTE 2 La partie à 4 bits de la SubClass de ce jeton n'est pas chiffrée. Les 57 bits restants sont chiffrés. NOTE 3 Le calcul du MAC n'inclut pas le FunctionIndex pour cette valeur de SubClass.				

Les types de services pris en charge sont indiqués dans le Tableau 19.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Tableau 19 – Types de service

ServiceType	Description	Notes
0	Arm load control switch (LCS) (armement de l'interrupteur de la charge)	Voir 6.2.5.2.3
1	Reset PIN (réinitialisation du PIN)	Voir 6.2.5.2.4.
2	Engineering mode (mode d'ingénierie)	Voir 6.2.5.2.5.
3	Disconnect LCS (déconnexion du LCS)	Voir 6.2.5.2.6
4	Reset tamper and arm LCS (réinitialisation de l'état de fraude et armement du LCS)	Voir 6.2.5.2.7.
5	Connect LCS (connexion du LCS (s'il est à l'état armé))	Voir 6.2.5.2.8.
6	Reset allowed overload count (réinitialisation du comptage des surcharges)	Voir 6.2.5.2.9
7	Disconnect LCS and generate balance settlement token (déconnexion du LCS et génération du jeton de remboursement)	Voir 6.2.5.2.10
8 à 31	Réservés	

6.2.5.2.2 ServiceType

Ce champ de données est une énumération simple conçue pour déclencher une fonction ou un comportement prédéfini, décrit dans le Tableau 19. La prise en charge de ces jetons de service supplémentaires doit être facultative au sein de la norme plus large, mais peut être obligatoire dans les spécifications d'accompagnement. Si un type de jeton particulier n'est pas pris en charge, le compteur doit alors renvoyer l'erreur de vérification 8 – FunctionError.

6.2.5.2.3 Arm load control switch (LCS)

Ce jeton ordonne au compteur de changer l'état de l'interrupteur de la charge (LCS) de Disconnected à Armed. A l'état Armed, le consommateur peut ensuite mener une action supplémentaire, non définie dans le présent document, pour reconnecter le LCS.

NOTE Ce jeton est destiné à être par exemple utilisé dans une situation où un compteur a reçu une instruction du WAN de déconnexion du LCS et le WAN est ensuite devenu indisponible, en laissant le consommateur sans alimentation. Le LCS peut être reconnecté en utilisant ce jeton, avec ou sans communication entre le WAN et le compteur.

6.2.5.2.4 Reset PIN

Ce jeton permet à un consommateur de réinitialiser le PIN de son compteur à une valeur par défaut (par exemple 0000).

NOTE Le consommateur peut ensuite choisir un autre PIN; ce choix ne relève pas du domaine d'application du présent document.

Les détails de la gestion du PIN doivent être convenus entre le vendeur du compteur et l'entreprise de distribution.

6.2.5.2.5 Engineering mode

Ce jeton permet d'activer un mode d'ingénierie du compteur pendant une durée limitée, pour exécuter des fonctions qui ne sont pas normalement présentées au consommateur (opérations d'installation et affichages supplémentaires, par exemple).

6.2.5.2.6 Disconnect LCS

Ce jeton permet à l'entreprise de distribution de déconnecter le LCS (par exemple à titre de sanction en cas de fraude).

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.2.5.2.7 Reset tamper and arm LCS

Ce jeton permet à l'entreprise de distribution de réinitialiser les indicateurs de fraude du compteur et de mettre le LCS à l'état armé. Le consommateur peut ensuite mener une action, non définie dans le présent document, pour reconnecter l'alimentation.

Les détails de l'action du consommateur doivent être convenus entre le vendeur du compteur et l'entreprise de distribution.

6.2.5.2.8 Connect LCS

Ce jeton connecte l'alimentation en utilisant le LCS. Il n'a aucun effet sauf si le LCS est déjà à l'état armé.

NOTE Il est destiné à être utilisé en situation d'urgence où l'interaction normale avec le consommateur n'est pas possible, par exemple lorsque le dispositif nécessaire d'introduction du consommateur (clavier par exemple) est défectueux. Dans ces circonstances, le jeton peut être envoyé par le biais du WAN pour reconnecter l'alimentation sans passer par un site.

6.2.5.2.9 Reset allowed overload count

Le compteur peut répondre à une condition de surcharge en utilisant l'interrupteur de la charge pour déconnecter l'alimentation, après quoi le consommateur peut réaliser une action pour reconnecter l'alimentation. Le nombre de ces cycles de déconnexion-reconnexion peut être limité (par exemple selon un plafond journalier) afin de s'assurer que la durée de vie du compteur n'est pas compromise.

Ce jeton permet à l'entreprise de distribution d'annuler une telle limite en réinitialisant le comptage des cycles de déconnexion-reconnexion afin que le consommateur puisse de nouveau reconnecter l'alimentation.

6.2.5.2.10 Disconnect LCS and generate balance settlement token

Ce jeton peut être utilisé pour un changement de locataire. Il permet à l'entreprise de distribution d'amener le compteur à déconnecter le LCS et générer un jeton codant et authentifiant le solde du compte.

Le jeton résultant généré par le compteur, dû à ce jeton de fonction spécial, est de SubClass 11 de la Class 5 et est décrit en 6.2.5.4. Il permet au client de rapporter, de manière sécurisée, à l'entreprise de distribution le solde de clôture du compte afin de faciliter le remboursement du solde créditeur dans le compteur.

6.2.5.3 SubClass 10: Jeton ExtendedTransferCredit

La structure générique du jeton de SubClass 10 de Class 5 est indiquée dans le Tableau 20.

Tableau 20 – Bloc 1 du jeton TransferCredit + Function

SubClass	TSTN	Données		TMAC
		AMTConfig	AMT	
4 bits	10 bits	2 bits	13 bits	32 bits
10 = 1010b Voir 6.2.3	Voir NOTE 2			Voir NOTE 2
NOTE 1 Les valeurs valides des champs TSTN, AMTConfig, AMT et TMAC et leurs interprétations sont identiques à celles des champs correspondants du jeton de SubClass 8 de Class 5 (voir 6.2.5.1). NOTE 2 Le champ TMAC est tronqué à partir des 128 bits du GMAC calculé sur tous les blocs participants. La méthode de calcul du GMAC à 128 bits est détaillée en 6.1.13, 6.1.15 et 6.1.16. Les jetons suivants dans le groupe de jetons ont une structure différente et doivent être saisis dans le compteur en ordre séquentiel afin que le processus d'application du compteur puisse réassembler la totalité du SuperToken.				

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Pour les jetons de SubClass 10 de la Class 5, les 4 bits MS du premier bloc (la SubClass) doivent être transmis non chiffrés et le bit MS doit être utilisé pour déterminer si tous les blocs du jeton sont chiffrés ou non.

L'ordre des jetons dans le groupe est étiqueté dans la structure des jetons suivants au moyen d'un compteur dans les 2 premiers bits du jeton. Le compteur indique le nombre de jetons supplémentaires à suivre (voir Tableau 21) après le jeton final du groupe ou le SuperToken ayant la BlockSequence = 0 conformément au Tableau 22.

Le CheckDigit dans le premier jeton s'applique aux 19 premiers chiffres du premier jeton dans le groupe uniquement, alors que les jetons suivants du groupe ont des chiffres de vérification qui sont calculés comme suit.

Pour le $N^{ème}$ jeton T_N (N > 1), ajouter le chiffre de vérification C_{N-1} du jeton précédent T_{N-1} aux 19 chiffres de T_N et calculer le chiffre de vérification des 20 chiffres résultants $C_{N-1} \parallel T_N$.

L'exemple suivant décrit l'ajout de données permettant de calculer le CheckDigit du second bloc de la TCDU à 40 chiffres:

- TCDU à 40 chiffres: 8889793723820927018101660992186693955792.
- Pour les 20 premiers chiffres de la TCDU, le CheckDigit associé est à obtenir à partir des 19 premiers chiffres (c'est-à-dire "8889793723820927018") en utilisant l'algorithme de Verhoeff et la valeur de CheckDigit "1".
- Pour le second bloc de la TCDU, le CheckDigit associé est à obtenir à partir du "1" (c'est-à-dire le 20e chiffre), ajouté au second bloc de 19 chiffres ("0166099218669395579") pour obtenir un total de 20 chiffres ("10166099218669395579") en utilisant l'algorithme de Verhoeff et la valeur de CheckDigit "2".

Le résultat du calcul est le CheckDigit C_N à ajouter à T_N.

Tableau 21 – Bloc 2 à *N-1* du jeton TransferCredit + Function *N* (*N* > 2)

BlockSequence	Données
2 bits	59 bits
01b – signifie qu'un seul bloc supplémentaire suit 10b – signifie que deux blocs supplémentaires suivent 11b – réservé Voir B.1.2	La signification de la sémantique spécifique à l'application sera traitée dans les spécifications d'accompagnement.

Tableau 22 – Dernier bloc du jeton TransferCredit + Function

BlockSequence	Données
2 bits	59 bits
= 00b Voir B.1.2	La signification de la sémantique spécifique à l'application sera traitée dans les spécifications d'accompagnement.

Pour un exemple de la façon dont les jetons séquentiels sont construits, voir l'Annex B.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.2.5.4 SubClass 11: jeton chiffré généré par le compteur

6.2.5.4.1 Généralités

Ce jeton est généré par le compteur dans le but de fournir à un POS ou HES un message sécurisé afin de rembourser au client un crédit inutilisé.

Bloc 1: jeton généré par le compteur pour ReturnedMeterData – la structure de jeton du bloc 1 pour la SubClass 11 de Class 5 doit être conforme au Tableau 23.

Tableau 23 – Bloc 1 pour la structure du jeton de SubClass 11 de Class 5 généré par le compteur

SubClass	TSTN	Données		TMAC
		Type de données	RES	
Binaire	Binaire	Binaire	Binaire	Binaire
4 bits	10 bits	5 bits	10 bits	32 bits
11d =1011b – Code généré par le compteur Voir 6.2.3	Numéro à 32 chiffres tronqué	Valeurs valides: 0d – Données de remboursement présentes dans le bloc 2 1d – 31d **Réservés pour une future utilisation**	**Réservé pour une future utilisation** Les bits réservés doivent être à zéro	Tronqué à partir des 128 bits du MAC (calculé sur tous les blocs participants)
NOTE Pour la SubClass 11 de la Class 5, le TSTN est le STN généré par compteur tronqué.				

De manière analogue aux autres sous-classes, le STN est maintenu par l'émetteur du jeton (le compteur dans ce cas) et ce dernier dérive le TSTN pour placer la valeur du TSTN dans le codage du bloc 1 pour cette SubClass.

Le MAC ne doit pas inclure le FunctionIndex pour cette valeur de SubClass.

La structure de jeton du bloc 2 pour la SubClass 11 de Class 5 est indiquée dans le Tableau 24.

Tableau 24 – Bloc 2 pour la structure du jeton de SubClass 11 de Class 5 généré par le compteur

Séquence de blocs	Mois et année de génération du jeton	Jour du mois de génération du jeton	Santé du compteur	Signe du montant de rapprochement	Configuration du montant de rapprochement	Montant de rapprochement	RES
Binaire	Binaire	Binaire	Binaire	Binaire	Binaire	Binaire	Binaire
2 bits	9 bits	5 bits	1 bit	1 bit	3 bits	18 bits	22 bits
valeur: 00b Voir B.1.2	Voir 6.2.5.4.2	Voir 6.2.5.4.3	Voir 6.2.5.4.4	Voir 6.2.5.4.5.	Voir 6.2.5.4.6	Voir 6.2.5.4.7	Réservé à un usage ultérieur.

6.2.5.4.2 Mois et année de génération du jeton

Le mois et l'année de génération du jeton sont représentés par une valeur de 9 bits variant de 0d à 511d. Elle décrit le numéro de mois depuis l'époque et la plage est suffisante pendant 41 ans à compter de l'époque. L'époque doit être spécifique au pays.

Par exemple: Si l'époque est Jan 2010, alors la valeur de 1d est Fev 2010.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.2.5.4.3 Jour du mois de génération du jeton

Valeurs valides: 0d (le 1er du mois) à 30d (le 31 du mois).

Les valeurs 28d à 30d représentant les 29e à 31e jours du mois ne sont pas valides si le mois correspond à ce nombre ou à un nombre de jours inférieur.

Par exemple: Si le mois est février, alors le 30 et le 31 sont des valeurs invalides. Si le mois est avril, alors le 31 est une valeur invalide.

6.2.5.4.4 Santé du compteur

Les valeurs suivantes sont valides:

- 0d – Fonctionnement normal;
- 1d – Fonctionnement anormal: par exemple, une condition de fraude persistante ou une défaillance matérielle détectée.

Les conditions à prendre en compte pour la santé du compteur sont spécifiques à chaque dispositif.

La signification de ces valeurs doit être convenue entre le vendeur du compteur et l'entreprise de distribution.

6.2.5.4.5 Signe du montant de rapprochement

Les valeurs suivantes sont valides:

- 0d – Le montant de rapprochement est positif: il est nécessaire que l'entreprise de distribution règle cette somme au consommateur;
- 1d – Le montant de rapprochement est négatif: il est nécessaire que l'entreprise de distribution recouvre cette somme auprès du consommateur.

NOTE Le terme "consommateur" désigne le consommateur qui occupe les locaux où le compteur a généré le jeton au moment de la génération des jetons.

6.2.5.4.6 Configuration du montant de rapprochement

Les valeurs suivantes sont valides:

- 0d – Signifie que le montant de rapprochement est en 1/1 000e de l'unité monétaire mineure du pays;
- 1d – Signifie que le montant de rapprochement est en 1/100e de l'unité monétaire mineure du pays;
- 2d – Signifie que le montant de rapprochement est dans l'unité monétaire mineure du pays;
- 3d – Signifie que le montant de rapprochement est en 10 fois l'unité monétaire mineure du pays;
- 4d – Signifie que le montant de rapprochement est en 100 fois l'unité monétaire mineure du pays;
- 5d – Signifie que le montant de rapprochement est en 1 000 fois l'unité monétaire mineure du pays;
- 6d – Signifie que le montant de rapprochement est en 10 000 fois l'unité monétaire mineure du pays;
- 7d – Réservé pour une future utilisation.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

6.2.5.4.7 Montant de rapprochement

Ce nombre qui, lorsqu'il est converti conformément au 6.2.5.4.6, indique le montant en unités monétaires majeures.

6.2.5.5 SubClass 12: RESERVEE

Cette SubClass est réservée pour une future affectation par le WG15 de l'IEC/TC13 et ne peut pas être utilisée à une autre fin.

6.2.5.6 SubClass 13: RESERVEE

Cette SubClass est réservée pour une future affectation par le WG15 de l'IEC/TC13 et ne peut pas être utilisée à une autre fin.

6.2.5.7 SubClass 14: RESERVEE

Cette SubClass est réservée pour une future affectation par le WG15 de l'IEC/TC13 et ne peut pas être utilisée à une autre fin.

6.2.5.8 SubClass 15: RESERVEE

Cette SubClass est réservée pour une future affectation par le WG15 de l'IEC/TC13 et ne peut pas être utilisée à une autre fin.

6.3 Eléments de données des jetons

Les éléments de données des jetons sont indiqués dans le Tableau 25.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Tableau 25 – Eléments de données des jetons

Nom de l'élément de données	Contexte	Format de l'élément de données	Taille	Plage	Référence
SubClass	Définit le type de jeton et si le jeton est chiffré ou non. Ces 4 bits ne sont jamais chiffrés.	Binaire	4 bits	Toutes les valeurs possibles	6.2.3
TSTN	Le TokenReferenceNumber est le bit LSB 10 du STN à 32 bits.	Binaire	10 bits	Toutes les valeurs possibles	6.1.7
AMTConfig	Définit le multiplicateur appliqué au champ AMT.	Binaire	2 bits	Toutes les valeurs possibles	6.2.4.1, 6.2.5.1
AMT	Le champ Montant définit la partie de mantisse de la somme et est combiné avec le champ AMTConfig pour représenter la valeur pour le jeton.	Binaire	13 bits	Toutes les valeurs possibles	6.2.4.1, 6.2.5.1
TMAC	Les 32 bits de poids faible du MessageAuthenticationCode généré pour le jeton au niveau de l'APDU.	Binaire	32 bits	Toutes les valeurs possibles	6.1.13 6.1.15
CheckDigit	Chiffre de vérification ajouté à la TCDU à la dernière étape de la génération.	Décimal	1 chiffre	0-9	6.2.5.3
ServiceType	Utilisé dans les jetons où les fonctions définies sont initiées pour le jeton SpecialFunction.	Binaire	5 bits	Toutes les valeurs possibles	6.2.5.2.2
RES	Utilisé dans les jetons où les fonctions sont initiées pour la SubClass 9 et la SubClass 11. Ces bits sont actuellement réservés pour une future affectation par le WG15 de l'IEC/TC13 et ne doivent pas être utilisés à d'autres fins spécifiques au fabricant. Ces bits inutilisés doivent être fixés à 0.	Binaire	10 bits	0000000000b	6.2.5.2

6.4 Fonctions de génération de la TCDU

La TCDU est créée lorsque l'APDU est convertie à partir d'une charge utile binaire en un nombre décimal et décalée par rapport aux autres domaines de classes et le CheckDigit est également ajouté.

La Figure 11 et la Figure 12 indiquent la façon dont une APDU est transformée en TCDU pour des jetons non chiffrés et chiffrés respectivement.

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

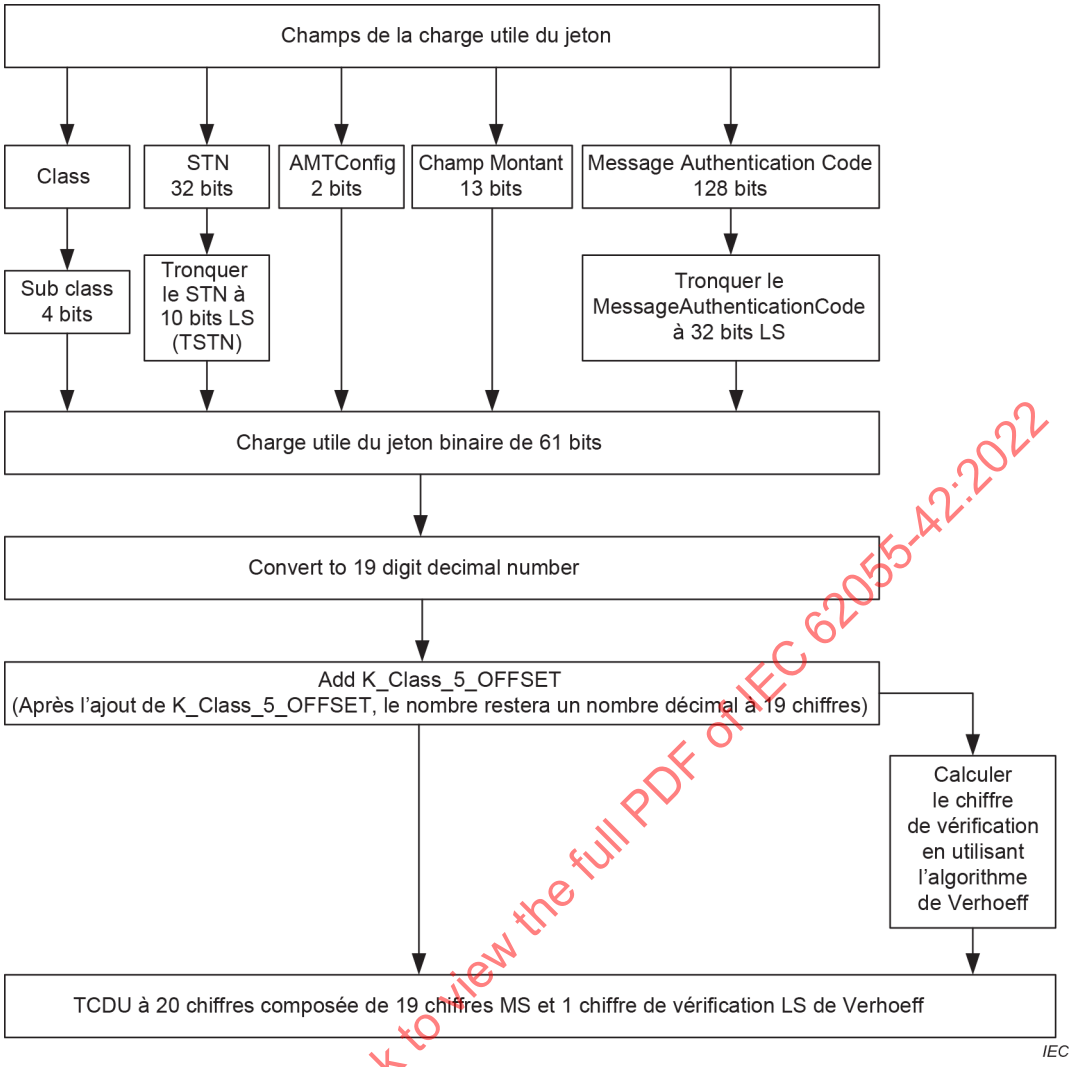
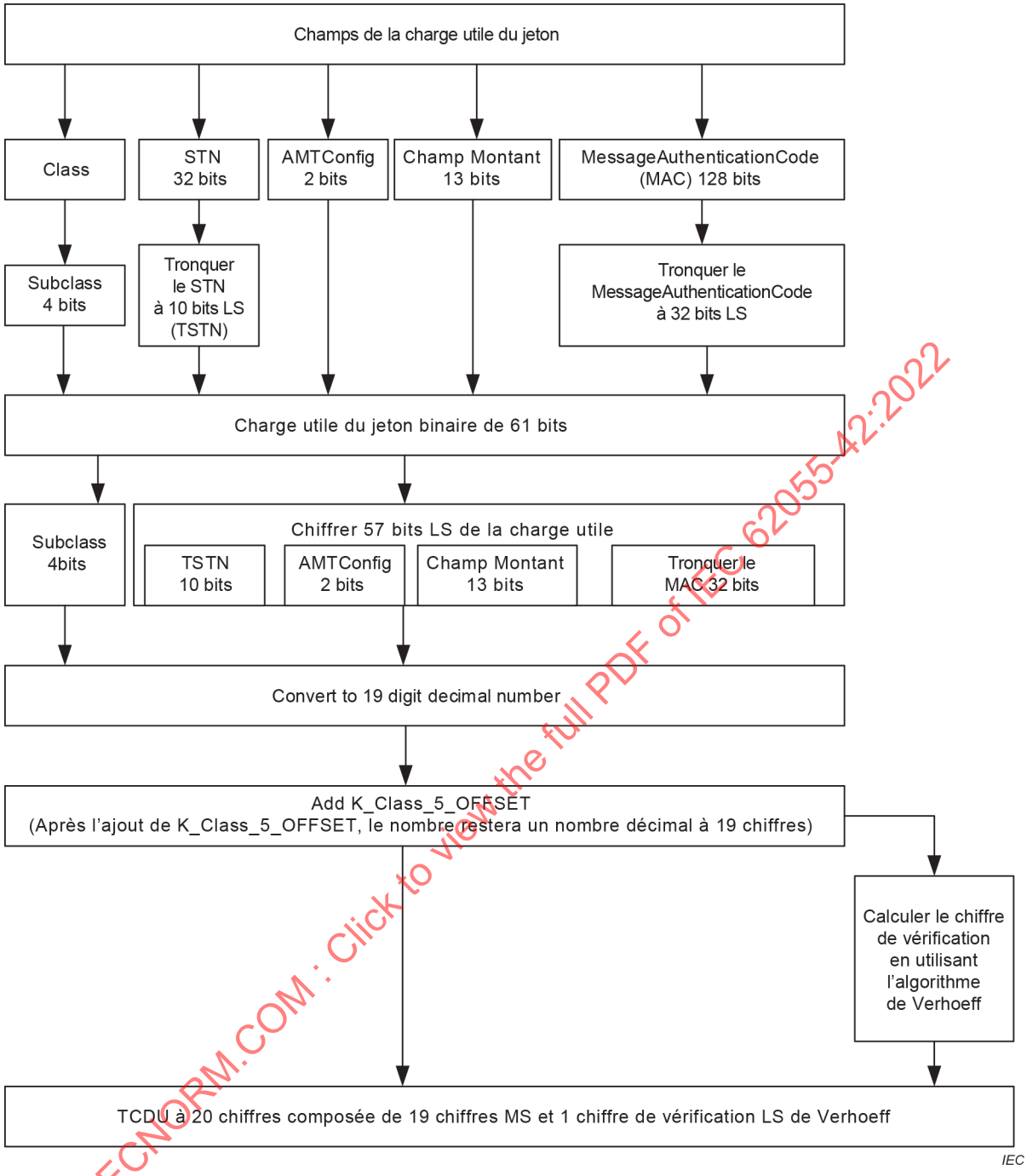


Figure 11 – Génération de la TCDU pour les jetons non chiffrés de SubClass 0

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022



IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

Figure 12 – Génération de la TCDU pour les jetons chiffrés de SubClass 8

Pour l'algorithme de chiffrement, se reporter au 6.5.4.

6.5 Fonctions de sécurité

6.5.1 Exigences générales

Le présent paragraphe décrit le mécanisme de chiffrement des jetons exigeant un chiffrement (SubClass 8-15), y compris le processus de génération et de gestion des clés, la politique de révision des clés et tous les autres aspects liés à la sécurité des jetons.

6.5.2 Gestion des clés

Les processus de gestion des clés ne sont pas définis dans le présent document.

6.5.3 Dérivation de clé

Les processus de dérivation de clé ne sont pas définis dans le présent document.

6.5.4 Processus de chiffrement

Un algorithme adapté de chiffrement préservant le format doit être utilisé. L'algorithme préservant le format n'est pas défini dans le présent document.

7 Protocole de couche application TokenCarriertoMeterInterface

7.1 APDU: ApplicationProtocolDataUnit

7.1.1 Eléments de données dans l'APDU

L'APDU est l'interface de données entre le MeterApplicationProcess et le protocole de couche application et comprend les éléments de données indiqués dans le Tableau 26.

Tableau 26 – Eléments de données dans l'APDU

Nom de l'élément de données	Contexte	Format de l'élément de données	Référence
Jeton	La charge utile du jeton est de 61 bits. Les 4 bits de poids fort définissent le type de jeton et si le jeton est chiffré ou non. Ces 4 bits ne sont jamais chiffrés.	61 bits	6.2.4.1 6.2.5.1 6.2.5.2
CheckDigit	Chiffre de vérification calculé sur la charge utile de l'APDU pour aider l'utilisateur en détectant la plupart des erreurs de saisie.	1 chiffre	5.3 6.2.5.3 Annex A
AuthenticationResult	Indicateur d'état du MeterApplicationProcess permettant de transférer le résultat provenant des contrôles d'authentification initiaux.		7.1.3
ValidationResult	Indicateur d'état du MeterApplicationProcess permettant de transférer le résultat provenant des contrôles de validation initiaux.		7.1.4
TokenResult	Indicateur d'état issu du MeterApplicationProcess permettant de transférer après traitement du jeton afin que le protocole de couche application puisse réaliser l'action appropriée.		7.1.5

IECNORM.COM : Click to view the full PDF of IEC 62055-42:2022

7.1.2 TokenData

Les TokenData provenant de la TCDU après déchiffrement (si cela est pertinent) et traitement sont présentées au MeterApplicationProcess dans l'APDU.

Le jeton réel à 61 bits, 122 bits, 183 bits ou 244 bits (TMAC de 32 bits inclus) est celui initialement saisi dans l'APDU par le MeterApplicationProcess. Le MeterApplicationProcess peut maintenant poursuivre son traitement.

7.1.3 AuthenticationResult

Cet indicateur d'état indique au MeterApplicationProcess que les contrôles d'authentification initiaux (voir 7.3.3) ont réussi ou échoué, afin que le MeterApplicationProcess puisse générer une réponse appropriée. Les valeurs possibles sont indiquées dans le Tableau 27.