INTERNATIONAL STANDARD

ISO 10202-5

First edition
1998-07-15

Financial transaction cards — Security architecture of financial transaction systems using integrated circuit cards —

Part 5:
Use of algorithms

Cartes de transactions financières — Architecture de sécurité des systèmes de transactions financières utilisant des cartes à circuit intégré —

Partie 5: Utilisation des algorithmes

Reference numbe
(E)

Contents

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland
Internet iso@iso.ch

Printed in Switzerland

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

International Standard ISO 10202-5 was prepared by Technical Committee ISO/TC 68, *Banking, securities and other financial services*, SC 6, *Retail financial services*.

ISO 10202 consists of the following parts, under the general title *Financial transaction cards — Security architecture of financial transaction systems using integrated circuit cards*:

— *Part 1: Card life cycle*

— *Part 2: Transaction process*

— *Part 3: Cryptographic key relationships*

— *Part 4: Secure application modules*

— *Part 5: Use of algorithms*

— *Part 6: Cardholder verification*

— *Part 7: Key management*

— *Part 8: General principles and overview*

Annexes A to H of this part of ISO 10202 are for information only.

Introduction

The standardisation of the security architecture of financial transaction systems using the Integrated Circuit Card (ICC) is partitioned into the following:

Part 1: Card life cycle

Part 2: Transaction process

Part 3: Cryptographic key relationships

Part 4: Secure application modules

Part 5: Use of algorithms

Part 6: Cardholder verification

Part 7: Key management

Part 8: General principles and overview

This part of ISO 10202 describes the cryptographic processes available to fulfill those security functions defined in parts 2, 4, and 6 where cryptographic algorithms are required.

ISO 10202 enables the execution of all security functions with either a symmetric or an asymmetric algorithm type. ISO 10202 does not cover zero-knowledge techniques, which could be incorporated at a later stage.

The nodes participating in a given cryptographic process shall each be capable of the cryptographic functionality required.

The cryptographic processes necessary to execute a security function are specified in options. Within each cryptographic process, a separate option is specified for each algorithm type. Also a separate option is specified for each variant of a cryptographic process that requires additional communication steps.

Clause 5 maps the security functions into the cryptographic processes available to achieve them.

Clause 6 specifies the details of the cryptographic processes. This part of ISO 10202 is not an implementation specification, but it does indicate those data elements required at both nodes to ensure that the cryptographic process is carried out in accordance with the required security procedures.

Financial transaction cards — Security architecture of financial transaction systems using integrated circuit cards —

Part 5:
Use of algorithms

1 Scope

This part of ISO 10202 applies to cryptographic exchanges where at least one node is an ICC or a SAM. Exchanges between other system nodes are outside the scope of this part of ISO 10202.

The provision of any security function is optional depending upon requirements of the system. Where a specific function is identified as being required, it shall be performed in the manner described herein.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 10202. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 10202 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 4909, *Bank cards — Magnetic stripe data content for track 3.*

ISO 9564-1, *Banking — Personal Identification Number management and security — Part 1: PIN protection principles and techniques*.

ISO/IEC 9796, *Information technology — Security techniques — Digital signature scheme giving message recovery*

Financial transaction cards — Security architecture of financial transaction systems using integrated circuit cards — Part 1: Card life cycle.

ISO 10202-2, *Financial transaction cards — Security architecture of financial transaction systems using integrated circuit cards — Part 2: Transaction process*.

ISO 10202-3, *Financial transaction cards — Security architecture of financial transaction systems using integrated circuit cards — Part 3: Cryptographic key relationships*.

ISO 10202-4, *Financial transaction cards — Security architecture of financial transaction systems using integrated circuit cards — Part 4: Secure application modules*.

ISO 10202-6, *Financial transaction cards — Security architecture of financial transaction systems using integrated circuit cards — Part 6: Cardholder verification*.

ISO 10202-7, *Financial transaction cards — Security architecture of financial transaction systems using integrated circuit cards — Part 7: Key management*.

ISO 10202-8, *Financial transaction cards — Security architecture of financial transaction systems using integrated circuit cards — Part 8: General principles and overview*.

3 Definitions

For the purposes of this part of ISO 10202, the following definitions apply.

3.1
asymmetric algorithm
an algorithm in which the encipherment and decipherment keys are different and for which it is computationally infeasible to deduce one from the other

3.2
certificate
(See public key certificate.)

3.3
certificate identifier
certificate information which enables proper verification of a key certificate

3.4
ciphertext
enciphered plaintext

3.5
collision resistant
a function is called collision resistant if any two different input values produce different output results

3.6
credentials
the set of data items assigned to each entity and used to authenticate that entity

3.7
cryptographic link
two logical entities (nodes) who have previously agreed to exchange data and who have a cryptographic key-relationship

3.8
cryptographic node
one of the logical entities (nodes) in a cryptographic link

3.9
decipherment
the process of transforming ciphertext into plaintext

3.10
digital signature
the result of a cryptographic transformation, executed by the initiator using his secret key with an asymmetric algorithm, providing non repudiation of the source and integrity of the signed data

3.11
distinguishing name
a name which uniquely identifies an entity in a process

3.12
encipherment
the process of transforming plaintext into ciphertext

3.13
entity authentication
corroboration that the identity of an entity (node) is the one claimed

3.14
explicit key identifier
(See key identifier.)

3.15
hash-function
a function which maps a string of bits to fixed-length strings of bits, satisfying the following two properties:

— it is computationally infeasible to find for a given output an input which maps to this output

— it is computationally infeasible to find for a given input a second input which maps to the same output

NOTE 1 The literature of the subject contains a variety of terms which have the same or similar meaning as hash-function. Compressed encoding and condensing function are some examples.

NOTE 2 Computational feasibility depends on the user specific security requirements and environment.

3.16
initiator
the node or entity that initiates a process

3.17
key
a parameter used in conjunction with a cryptographic algorithm for executing cryptographic transformations

3.18
key identifier
key information which enables the recipient to determine the appropriate keys(s) associated with a transaction

3.19
message authentication
process providing cryptographic proof that a message has not been altered or destroyed in an unauthorized manner

3.20
message authentication code (MAC)
a data field, the contents of which can be used to verify the integrity of a message, or selected message elements

3.21
non-repudiation
security service providing permanent cryptographic proof of the integrity and origin of data — both in an unforgeable relationship — which can be verified by any third party at any time

3.21
one-way function
a mathematical function which maps input values into output values in an irreversible way

3.22
plaintext
intelligible data that has a meaning and can be read or acted upon without the application of a transformation

3.23
public key certificate (certificate)
a set consisting of user credentials (including the public key) together with the trusted third party's digital signature of these credentials

3.24
reflection attack
an attack by a false respondent whereby the initiator is challenged in a separate session with the same random value it has transmitted in a concurrent session to authenticate that respondent

3.25
respondent
the node or entity that responds to the initiator of a process

3.26
symmetric algorithm
a cryptographic method using the same secret key for encipherment and decipherment

3.27
timeliness
a method to prevent a valid message from being replayed at a later time e.g. by using a probe of information as a challenge requesting a proper and timely response

3.28
token
a set of data items formed for each data exchange sent by one entity to another

3.29
transaction certification code
result of the transaction certification process producing an electronic signature, which could either be a MAC (based on a symmetric algorithm) or a digital signature (based on an asymmetric algorithm)

3.30
transaction certification
process providing cryptographic proof of the origin and integrity of transaction data which can be verified by a third party

3.31
trusted third party
a generally accessible entity being known and trusted by the communicating entities

4 Notations

Throughout this part of ISO 10202 the following notations are used.

4.1 Values and entities

Values and entities are in italic print:

A	the distinguishing name of entity *A*.
Cert$_X$	certificate of entity *X*.
CID$_X$	certificate identifier of entity *X* (see annex B).
Cred$_X$	credential of entity *X* (see annex A).
k$_X$	a key (K_X, S_X, P_X) associated with entity *X*.
K$_X$	a secret key associated with entity *X* to be used with a symmetric algorithm.
KID$_{kX}$	explicit key identifier for key *k* of entity *X* (see annex B).
KID$_{PX}$	explicit key identifier for key pair S_X/P_X of entity *X* (see annex B).
PBF0	PIN block format 0 according to ISO 9564.
PBF1	PIN block format 1 according to ISO 9564.
R$_X$	random value issued by entity *X*.
S$_X$/P$_X$	a public/secret key associated with entity *X* to be used with an asymmetric algorithm.
TP	the distinguishing name of the trusted third party.
T$_{val}$	validity period for credentials.
T$_X$	time stamp issued by entity *X*.
Z||Z*	the concatenation of the bitstrings *Z* and *Z**.
<>|<>	separation of fields

4.2 Processes

Process identifiers are in upper case:

EA	entity authentication
KE	key exchange
MA	message authentication
ME	message encipherment/decipherment
PV	PIN verification
TC	transaction certification

4.3 Optionlist

Option identifiers are in lower case:

a	asymmetric
s	symmetric
m	mutual
t	timeliness

4.4 Functions

Function identifiers are in lower case italic print:

c	compare
d	decipher
e	encipher
g	generate a random
h	hash
m	authenticate message
o	apply a one-way-function
s	sign
v	verify

The function notation is used in combination with an indication of the key values and entities.

c(Y,Z)	the comparison of two bitstrings *Y* and *Z*; result is a status code.
dK(Z)	the decipherment of data *Z* by a symmetric algorithm using key *K*.
dS$_X$(Z)	the decipherment of data *Z* by an asymmetric algorithm using the secret key S_X.
eK(Z)	the encipherment of data *Z* by a symmetric algorithm using key *K*.
eP$_X$(Z)	the encipherment of data *Z* by an asymmetric algorithm using the public key P_X.
R=g()	the generation of a random value *R*.
h(Z)	the application of a collision-resistent one-way function which maps a data item *Z* to an output value of a fixed length using public properties (hashing); the hash result is *h(Z)*.
mK(Z)	the generation of a message authentication code (MAC-ing) using a symmetric algorithm as a one-way function with the secret key *K*; the result is a Message Authentication Code MAC.
vK(MAC)	the verification of a MAC using a symmetric algorithm as a one-way function with the secret key *K*; the result is a status code.
oK(Z)	the application of a one-way function which maps data Z to an output value of a fixed length using an algorithm with a secret key *K*.
sS$_X$(Z)	the application of a digital signature to data *Z* using the secret key S_X (signing); the result is a signature *Sig.*
vP$_X$(Sig)	the verification process of *Sig* using the public key P_X; the result is a status code.

4.5 Digital signatures

The transmission of the unsigned *Z* (data to be transferred) is mandatory except when a digital signature scheme with message recovery is used (see ISO/IEC 9796). In this case the data signed is composed as such that its structural properties can be verified and that *Z* can be retrieved from it. This requires *Z* to be sufficiently short and the asymmetric algorithm to be reversible.

Throughout this part of ISO 10202 the notation *sS$_A$(Z)* is used for both signing with data recovery or signing using a hash function. This means that *sS$_A$(Z)* can either stand for *sS$_A$(Z)* or *sS$_A$(h(Z))//Z* where *h(Z)* could denote *Z*.

4.6 Security message format

Security messages are logical commands for standardised security functions. The information of these security messages can be transmitted in the ICC related field of an 8583 message or interpreted by a SAM or application to produce the appropriate commands for ICCs.

A security message is identified as follows:

- **message indicator** logical message identifier; a concatenation of the following elements:

  process: process identifier (subclause 4.2)

  optionlist: list of option identifiers (subclause 4.3)

  number: sequential message number

EXAMPLE KEss1

The security message contains one or more security message subfields. Mandatory subfields are in boldprint. The <operation> subfield appears at least once in every message.

- **message subfields** *<initiator>* | *<respondent>* | *<operation>* | ***<operation>***

  <initiator>: distinguishing name of source entity

  : distinguishing name of destination entity

  <operation>= *<Z>* | *<KID>* | *<**f(Z)**>*

  <Z> : optional datafield

  <KID>: key identifier

  <f(Z)>: result of applying the function *f* to the data *Z* (subclause 4.4)

EXAMPLE *A* | *B* | KID_K | KID_{K^*} | $eK^*(KID_{K^*})$ | $eK(K^*)$

5 Mapping security functions to process types

A mapping of the security functions as defined in parts 2, 4 and 6 of ISO 10202 and the set of process types is shown in table 1.

<respondent>

Table 1 — Mapping of security functions to process types

SECURITY FUNCTION	PROCESS TYPE		KEY
ICC manufacturing			
Manufacturing	Initial key loading		$kMprd_{M\text{-}P}$
Embedding and initialisation	Initial key loading		$kEprd_{E\text{-}P}$
ICC personalisation			
Personalisation	Initial key loading		$kIctl_{I\text{-}C}$
Card session initialisation			
IC compability check	No cryptographic process		
CDF parameters update			
CDF activation/deactivation/reactivation/termination	Key Exchange	KE	$kIctl_{I\text{-}C}$
ADF allocation	Key Exchange	KE	$kIctl_{I\text{-}C}$
CDF specific authentication and verification			
Cardholder verification	PIN verification	PV	$kIenc_{I\text{-}C}$[1]
CDF static authentication	Entity Authentication	EA	$kIaut_{I}$
CDF dynamic authentication	Entity Authentication	EA	$kIaut_{C}$
Issuer host dynamic authentication	Entity Authentication	EA	$kIaut_{I\text{-}C}$
CDF transaction handling			
Transaction authorisation	Message Authentication	MA	$kImac_{I\text{-}C}$
Data confidentiality	Message Encipherment	ME	$kIenc_{I\text{-}C}$[1]
Transaction certification	Transaction Certification	TC	$kIcer_{I\text{-}C}$
ADF selection			
ADF selection	No cryptographic process		
ADF parameters update			
ADF $KActl_{A\text{-}C}$ loading	Key Exchange	KE	$kIkex_{I\text{-}A}$
ADF activation/deactivation/reactivation/termination	Key Exchange	KE	$kActl_{A\text{-}C}$
ADF specific authentication and verification			
Cardholder verification	PIN verification	PV	$kAenc_{A\text{-}C}$[2]
ADF static authentication	Entity Authentication	EA	$kAaut_{A}$
ADF dynamic authentication	Entity Authentication	EA	$kAaut_{C}$
SAM authentication	Entity Authentication	EA	$kAaut_{S\text{-}C}$
Application supplier authentication	Entity Authentication	EA	$kAaut_{A\text{-}C}$
ADF transaction handling			
Transaction authorisation	Message Authentication	MA	$kAmac_{A\text{-}C}$
Data confidentiality	Message Encipherment	ME	$kAenc_{A\text{-}C}$[2]
Transaction certification	Transaction Certification	TC	$kAcer_{A\text{-}C}$
Card session termination			
Card session termination	No cryptographic process		

1 At the issuers discretion these keys can be separately generated or derived.
2 At the providers discretion these keys can be separately generated or derived.

6 Process specifications

This clause specifies the processes which are available for the different security functions. Each process may be used on its own or in combination with other processes as required.

Each process allows several options that protect against different threats like interception, masquerade, replay, manipulation or repudiation. The selection of options should be based on an analysis of the risk resulting from the threats in a specific environment. Further guidance on this clause can be taken from Annex H.

Some processes include options that provide for timeliness where it is required to ensure that a message is unique and/or sent at a specified time. Different ways to obtain timeliness are explained in Annex E.

For Entity Authentication the timeliness is inherent; combination with such a process automatically ensures timeliness.

If the cryptographic process is based on a symmetric algorithm, the secret key required for the process shall have been established at both participating nodes. If the secret key is shared between more than two entities belonging to a group it is cryptographically impossible to distinguish between those entities.

If the cryptographic process is based on an asymmetric algorithm, key information shall be generated at each node. The secret key shall be stored in a secure manner at the node, and the corresponding public information shall be certified by a trusted third party that provides certificates (see Annex A).

All data elements in a message have to be known at the destination node in order to be able to execute the next step in the process. If a data element is generated dynamically during the process, its transmission is mandatory. If a data element is already known at the destination node, its transmission may be omitted.

Operations and parameters in boldprint are mandatory. All mandatory elements are reflected in the arrows of the figures. Alternative steps described in the roles may not always correspond directly to the steps described in that figure.

6.1 Process 1: Key Exchange (KE)

The secure electronic transmission of a secret key K or S_B to or from an Integrated Circuit Card or a SAM shall be carried out according to one of the options described in this subclause. It is assumed that the initiator and respondent have established a cryptographic link.

The secret key which is to be exchanged is always sent from *A* to *B*, except where the key is mutually developed by the two communicating parties.

6.1.1 KE-symmetric-symmetric

Exchange of a secret key K^* to be used with a symmetric algorithm by means of a symmetric algorithm (figure 1).

KEss1 = $A \mid B \mid KID_K \mid KID_{K^*} \mid eK^*(KID_{K^*}) \mid$ ***eK(K*)***

K	common secret key
K^*	exchanged secret key
KID_K	key identifier for key K
KID_{K^*}	key identifier for key K^*