



BSI Standards Publication

Identification cards — Integrated circuit cards

Part 8: Commands and mechanisms for security operations

National foreword

This British Standard is the UK implementation of ISO/IEC 7816-8:2021+A1:2023. It supersedes BS ISO/IEC 7816-8:2021, which is withdrawn.

The start and finish of text introduced or altered by amendment is indicated in the text by tags. Tags indicating changes to ISO/IEC text carry the number of the ISO/IEC amendment. For example, text altered by ISO/IEC amendment 1 is indicated by **A1** **A1**.

The UK participation in its preparation was entrusted to Technical Committee IST/17, Cards and security devices for personal identification.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2024
Published by BSI Standards Limited 2024

ISBN 978 0 539 21098 9

ICS 35.240.15

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 August 2021.

Amendments/corrigenda issued since publication

Date	Text affected
31 March 2024	Implementation of ISO/IEC amendment 1:2023

Identification cards — Integrated circuit cards —

Part 8: Commands and mechanisms for security operations

Cartes d'identification — Cartes à circuit intégré —

Partie 8: Commandes et mécanismes pour les opérations de sécurité



Reference number
ISO/IEC 7816-8:2021(E)

AVA Documents
Document Sharing Hub
© ISO/IEC 2021



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2021

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier; Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword	iv
Introduction	v
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Abbreviated terms	4
5 Interindustry commands for security operations	5
5.1 General	5
5.2 GENERATE ASYMMETRIC KEY PAIR command	5
5.3 PERFORM SECURITY OPERATION command	25
5.3.1 General	25
5.3.2 COMPUTE CRYPTOGRAPHIC CHECKSUM operation	27
5.3.3 COMPUTE DIGITAL SIGNATURE operation	28
5.3.4 HASH operation	28
5.3.5 VERIFY CRYPTOGRAPHIC CHECKSUM operation	29
5.3.6 VERIFY DIGITAL SIGNATURE operation	29
5.3.7 VERIFY CERTIFICATE operation	30
5.3.8 ENCIPHER operation	31
5.3.9 DECIPHER operation	31
5.3.10 "QSC COMPUTE DIGITAL SIGNATURE operation"	31
5.3.11 "QSC VERIFY DIGITAL SIGNATURE operation"	33
5.3.12 "QSC usage of VERIFY CERTIFICATE operation"	34
5.3.13 "QSC ENCIPHER operation"	34
5.3.14 "QSC DECIPHER operation"	34
Annex A (informative) Examples of operations related to digital signature	36
Annex B (informative) Examples of certificates interpreted by the card	43
Annex C (informative) Examples of asymmetric key transfer	57
Annex D (informative) Alternatives to achieve the reversible change of security context	60
Annex E (informative) Examples of uses for GENERATE ASYMMETRIC KEY PAIR command	62
Annex F (informative) QSC Miscellaneous	68
Bibliography	79

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 17, *Cards and personal identification*.

This fifth edition cancels and replaces the fourth edition (ISO/IEC 7816-8:2019), which has been technically revised.

The main changes compared to the previous edition are as follows:

- in [Table A.9](#), [A.10](#) and [A.11](#), P1-P2 value of MSE command has been corrected;
- in [Table A.11](#), P1-P2 value of PSO command with HASH operation has been corrected.

A list of all parts in the ISO/IEC 7816 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

ISO/IEC 7816 is a series of standards specifying integrated circuit cards and the use of such cards for interchange. These cards are identification cards intended for information exchange negotiated between the outside world and the integrated circuit in the card. As a result of an information exchange, the card delivers information (computation result, stored data) and/or modifies its content (data storage, event memorization).

Five parts are specific to cards with galvanic contacts and three of them specify electrical interfaces:

- ISO/IEC 7816-1 specifies physical characteristics for cards with contacts;
- ISO/IEC 7816-2 specifies dimensions and location of the contacts;
- ISO/IEC 7816-3 specifies electrical interface and transmission protocols for asynchronous cards;
- ISO/IEC 7816-10 specifies electrical interface and answer to reset for synchronous cards;
- ISO/IEC 7816-12 specifies electrical interface and operating procedures for USB cards.

All the other parts are independent from the physical interface technology. They apply to cards accessed by contacts and/or by radio frequency:

- ISO/IEC 7816-4 specifies organization, security and commands for interchange;
- ISO/IEC 7816-5 specifies registration of application providers;
- ISO/IEC 7816-6 specifies interindustry data elements for interchange;
- ISO/IEC 7816-7 specifies commands for structured card query language;
- ISO/IEC 7816-8 specifies commands for security operations;
- ISO/IEC 7816-9 specifies commands for card management;
- ISO/IEC 7816-11 specifies personal verification through biometric methods;
- ISO/IEC 7816-13 specifies commands for handling the life cycle of applications;
- ISO/IEC 7816-15 specifies cryptographic information application.

ISO/IEC 10536 (all parts) specifies access by close coupling. ISO/IEC 14443 (all parts) and ISO/IEC 15693 (all parts) specify access by radio frequency. Such cards are also known as contactless cards.

Identification cards — Integrated circuit cards —

Part 8:

Commands and mechanisms for security operations

1 Scope

This document specifies interindustry commands which can be used for security operations. This document also provides informative directives on how to construct security mechanisms with commands defined in ISO/IEC 7816-4.

The choice and conditions of use of cryptographic mechanism in security operations can affect card exportability. The evaluation of the suitability of algorithms and protocols is outside the scope of this document. It does not cover the internal implementation within the card and/or the outside world.

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.

[A1] ISO/IEC 7816-4:2020 **[A1]**, *Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

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

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

3.1

asymmetric key pair

pair of elements belonging to cryptographic techniques that use two related operations: a public operation defined by public numbers or by a public key (3.4) and a private operation defined by private numbers or by a private key

Note 1 to entry: The two operations have the property that, given the public operation, it is computationally infeasible to derive the private operation.

3.2

certificate

digital signature (3.3) binding a particular person or object and its associated public key (3.4)

Note 1 to entry: The entity issuing the certificate also acts as tag allocation authority with respect to the data elements in the certificate.

[SOURCE: ISO/IEC 7816-4:2020, 3.11]

3.3

digital signature

data appended to, or cryptographic transformation of, a data string that proves the origin and the integrity of the data string and protects against forgery, e.g. by the recipient of the data string

[SOURCE: ISO/IEC 7816-4:2020, 3.20]

3.4

key

sequence of symbols controlling a cryptographic operation

EXAMPLE Encipherment, decipherment, a private or a public operation in a dynamic authentication, signature production, signature verification.

[SOURCE: ISO/IEC 7816-4:2020, 3.30]

3.5

non-self-descriptive certificate

certificate (3.2) consisting of a concatenation of data elements associated to a header list or extended header list, describing the structure of the certificate

3.6

self-descriptive certificate

certificate (3.2) consisting of a concatenation of data objects

3.7

secure messaging

SM

set of means for cryptographic protection of (parts of) command-response pairs

[SOURCE: ISO/IEC 7816-4:2020, 3.49]

A1 3.8

certificate chain

ordered list of certificates that starts with an end-entity certificate, includes one or more certificate authority (CA) certificates, and ends with the end-entity certificate's root CA certificate, where each certificate in the chain is the certificate of the CA that issued the previous certificate

3.9

classic McEliece

code-based quantum safe key encapsulation algorithm retained in the course of the third round of the National Institute of Standards and Technology (NIST) contest

3.10

code-based

cryptosystem based on error correcting code

3.11

common parameter

public value that is used to control the operation of a cryptographic algorithm or that is used by a cryptographic algorithm to compute outputs

3.12

crypto-agility

property that permits changing or upgrading cryptographic algorithms or parameters

[SOURCE: ETSI TR 103 619^[14]]

3.13

crystals-dilithium

lattice-based quantum safe signature algorithm as selected by the National Institute of Standards and Technology (NIST) contest for standardization **A1**

A1 3.15**discrete logarithm**

computation of logarithms with regards to multiplicative cyclic groups

Note 1 to entry: This problem is considered as a hard mathematical problem when the number is large, and is the basis of some asymmetric cryptography scheme.

3.16**El Gammal**

asymmetric encryption/decryption protocol based on the discrete logarithm problem and using a throwable mask

3.17**falcon**

lattice-based quantum safe signature algorithm as selected by the National Institute of Standards and Technology (NIST) contest for standardization

3.18**frodoKEM**

lattice-based quantum safe key encapsulation algorithm retained in the course of the third round of the National Institute of Standards and Technology (NIST) contest

3.19**hash-based**

digital signatures constructed using hash functions

3.20**hybrid certificate**

certificate secured using both a regular asymmetric signature algorithm and a quantum safe signature algorithm

3.21**key encapsulation algorithm**

class of encryption techniques designed to secure symmetric cryptographic key material for transmission using asymmetric (public-key) algorithms

Note 1 to entry: The term used by National Institute of Standards and Technology (NIST) is “Key Encapsulation Mechanism”.

3.22**lattice-based**

cryptosystem based on lattice problems

3.23**Leighton Micali signature****LMS**

hash-based quantum safe signature algorithm

Note 1 to entry: This algorithm is defined in Reference [15].

3.24**Merkle tree**

data structure where the data is hashed and combined until there is a singular root hash that represents the entire structure

3.25**NTRU**

lattice-based quantum safe key encapsulation algorithm retained in the course of the third round of the National Institute of Standards and Technology (NIST) contest **A1**

3.26

public key parameters

set of components and characteristics describing the public key

3.27

private key parameters

set of components and characteristics describing the private key

3.28

quantum safe

characteristic of an algorithm or cryptosystem that is secure against both quantum and classical computers

3.29

quantum safe cryptography

cryptographic systems that are secure against both quantum and classical computers

Note 1 to entry: This encompasses any cryptosystem (e.g. symmetric, asymmetric).

3.30

Saber

lattice-based quantum safe key encapsulation algorithm retained in the course of the third round of the National Institute of Standards and Technology (NIST) contest

3.31

sphincs+

hash-based quantum safe signature algorithm as selected by the National Institute of Standards and Technology (NIST) contest for standardization

Note 1 to entry: See Table AMD.1.20.

3.32

XMSS

hash-based quantum safe signature algorithm

Note 1 to entry: The algorithm is defined in Reference [15]. **A1**

4 Abbreviated terms

A1 AlgID Algorithm Identifier **A1**

BCD binary-coded decimal

BER basic encoding rules of ASN.1 (see ISO/IEC 8825-1)

CA certification authority

CCT control reference template for cryptographic checksum

CRT control reference template

A1 CRT(2) optimized mode of computation for RSA algorithm exploiting the properties of the Chinese remainder theorem, and making use of a private key under a reduced form **A1**

CT control reference template for confidentiality

CVCA country verifying CA

DG3 data group 3

DO BER-TLV data object

DO'...'	BER-TLV data object, the tag of which is a hexadecimal value given between single quo-tation marks
DSA	digital signature algorithm
DST	control reference template for digital signature
DV	document verifier
ECC	elliptic curve cryptography
ECDH	elliptic curve Diffie–Hellman key exchange
ECDSA	elliptic curve digital signature algorithm
EF	elementary file
GQ2	modified Guillou-Quisquater protocol for zero knowledge proof
 HSS	hierarchical signature system 
HT	control reference template for hash-code
ICAO	international civil aviation organization
ICC	integrated circuit card
IS	inspection system
KAT	control reference template for key agreement
LDS	logical data structure
 LMS	Hash-based Leighton Micali Signature
<i>LM-OTS</i>	Leighton Micali One-Time Signature 
MRTD	machine readable travel document
MSE	MANAGE SECURITY ENVIRONMENT command
OID	object identifier, as defined by ISO/IEC 8825-1
 PKI	Public Key Infrastructure 
PSO	PERFORM SECURITY OPERATION command
 QSC	Quantum Safe Cryptography 
RFU	reserved for future use for ISO/IEC JTC 1/SC 17
RSA	Rivest, Shamir, Adleman
SE	security environment
SEID	security environment identifier
 SFM	regular mode of computation of RSA algorithm 
TLV	tag, length, value
 XMSS	Extended Merkle (Hierarchical) Signature Scheme 

5 Interindustry commands for security operations

5.1 General

An ICC compliant with this document may support any of the commands and/or options provided in 5.2 and 5.3. ^{A1} This document aims additionally at providing a set of generic QSC features. These generic features cater to a variety of QSC algorithms. This approach seeks to enhance this document with crypto-agility means and hybrid cryptography capabilities. ^{A1}

NOTE In addition to the use of logical channels, there are other alternatives that can be used for switching the security context. [Annex D](#) provides information about this functionality.

5.2 GENERATE ASYMMETRIC KEY PAIR command

^{A1} 5.2.1 General ^{A1}

The GENERATE ASYMMETRIC KEY PAIR command, which shall be as specified in [Table 1](#), initiates

- the generation and storing of an asymmetric key pair, i.e. a public key and a private key, in the card,
- the generation, storing of an asymmetric key pair and extracting generated public key, or
- the extracting previously generated public key.

The command may be preceded by a MANAGE SECURITY ENVIRONMENT command in order to set key generation related parameters (e.g. algorithm reference). The command may be performed in one or several steps, possibly using command chaining (see ISO/IEC 7816-4).

Table 1 — GENERATE ASYMMETRIC KEY PAIR command-response pair

CLA	As defined in ISO/IEC 7816-4
INS	'46' or '47'
P1	See Table 2
P2	'00' (no information provided) or reference of the private key to be generated coded according to ISO/IEC 7816-4:2020, Table 102
L _c field	Absent for encoding N _c = 0, present for encoding N _c > 0
Data field	Absent, or Proprietary data if P1-P2 set to '0000', or One or more CRTs associated to the key generation if P1-P2 different from '0000' (see notes) A CRT may include an extended header list
L _e field	Absent for encoding N _e = 0, present for encoding N _e > 0
Data field	Absent, or Public key as a sequence of data elements (INS = '46'), or Public key as a sequence of data objects (INS = '47'), or Public key as a sequence of data objects according to an extended header list (INS = '47')
SW1-SW2	See ISO/IEC 7816-4:2020, Tables 6 and 7 where relevant, e.g. 6985

NOTE 1 Several CRTs are present when the key pair is generated for several uses. In the command data field, a CRT possibly has a zero length.

Table 2 — P1 coding

b8	b7	b6	b5	b4	b3	b2	b1	Value
0	0	0	0	0	0	0	0	No information given
1	—	—	—	—	x	x	x	Additional information given
1	—	—	—	—	—	—	x	Key generation
1	—	—	—	—	—	—	0	- Generate asymmetric key pair
1	—	—	—	—	—	—	1	- Access to an existing public key
1	—	—	—	—	—	x	—	Format of returned public key data
1	—	—	—	—	—	0	—	- Proprietary format
1	—	—	—	—	—	1	—	- Output format according to extended header list
1	—	—	—	—	x	—	—	Output indicator
1	—	—	—	—	0	—	—	- Public key data in response data field
1	—	—	—	—	1	—	—	- No response data if Le field absent or proprietary if Le field present
—	x	x	x	x	—	—	—	0000, other values are RFU

NOTE 2 The private key can be stored in

- an internal EF the reference of which is known before issuing the command, or
- a DO'7F48' as cardholder private key template, or
- a QSC template DO'7F75' featuring private key with private key parameters, or
- a QSC template DO'7F76' featuring private key with private key parameters alongside common parameters.

NOTE 3 The public key can be stored for example in

- a DO'7F49' as cardholder public key template, or
- a QSC template DO'7F75' featuring exclusively public key parameters, or
- a QSC template DO'7F76' featuring public key parameters alongside common parameters, see examples on Table AMD.1.2 to Table AMD.1.7.

For extracting a previously generated public key (i.e. no generation), the command data field shall be empty or shall contain a CRT, possibly including an extended header list.

NOTE 4 In those cases when only access to a previously generated public key is requested, P2 is either '00' or references the private key.

The response data field shall be

- absent, $\boxed{A_1}$ or $\langle A_1 \rangle$
- a public key as a sequence of data elements (INS = '46'), $\boxed{A_1}$ or $\langle A_1 \rangle$
- a public key as a sequence of data objects (INS = '47') from Table 3, or
- a public key as a DO'7F49' (INS = '47') nesting data objects from Table 3 $\boxed{A_1}$, or
- a QSC template DO'7F75' (INS = '47') encapsulating public key and QSC key component data objects from Table AMD.1.1, or
- a QSC template DO'7F76' (INS = '47') encapsulating public key and QSC common parameter component data objects from Table AMD.1.1.

EXAMPLE 1 The QSC templates are provided in Table AMD.1.2 and Table AMD.1.5 for the hash-based signature algorithm.

EXAMPLE 2 The QSC templates are provided in Table AMD.1.3, Table AMD.1.4, Table AMD.1.6 and Table AMD.1.7 for the lattice-based algorithm. $\langle A_1 \rangle$

If the command data field does not indicate any format of public key data, it shall be implicitly known before issuing the command (e.g. as part of the security environment). When the command data field indicates an extended header list within a CRT, it covers public key data objects and other requested data object.

EXAMPLE 3 [Annex E](#) provides a set of examples on the use of this command.

If the algorithm is not indicated in the command, then the algorithm is known before issuing the command. In the public key template, the context-specific class (first byte from '80' to 'BF') is reserved for public key data objects.

Table 3 — Public key data objects

Tag	Value
'7F49'	Interindustry template for nesting one set of public key data objects with the following tags
'06'	Object identifier of any further information, optional
'80'	Algorithm reference as used in control reference data objects for secure messaging, optional
	Set of public key data objects for RSA
'81'	Modulus (a number denoted as n coded on x bytes)
'82'	Public exponent (a number denoted as v , e.g. 65 537)
	Set of public key data objects for DSA
'81'	First prime (a number denoted as p coded on y bytes)
'82'	Second prime (a number denoted as q dividing $p-1$, e.g. 20 bytes)
'83'	Basis (a number denoted as g of order q coded on y bytes)
'84'	Public key (a number denoted as y equal to g to the power x mod p where x is the private key coded on y bytes)
	Set of public key data objects for ECC
'81'	Prime (a number denoted as p coded on z bytes)
'82'	First coefficient (a number denoted as a coded on z bytes)
'83'	Second coefficient (a number denoted as b coded on z bytes)
'84'	Generator (a point denoted as PB on the curve, coded on $2z + 1$ or $2z$ or $z + 1$ bytes)
'85'	Order (a prime number denoted as q , order of the generator PB , coded on z bytes)
'86'	Public key (a point denoted as PP on the curve, equal to x times PB where x is the private key, coded on $2z + 1$ or $2z$ or $z + 1$ bytes)
'87'	Co-factor
	Set of public key data objects for GQ2
'81'	Modulus (a number denoted as n coded on x bytes)
'83'	Number of basic numbers (a number denoted as m coded on 1 byte. If tag '83' is present, then tag 'A3' shall be absent and the m basic numbers denoted as $g, g_2..g_m$ are the first m prime numbers 2, 3, 5, 7, 11...)
'84'	Verification parameter (a number denoted as k coded on 1 byte)
'A3'	Set of m basic numbers denoted as $g, g_2..g_m$, each one coded on 1 byte with tag '80' (If tag 'A3' is present, then tag '83' shall be absent)

Table 3 (continued)

Tag	Value
	Set of public key data objects for RSA Okamoto-Schnorr signature scheme^a
'81'	p the first large prime number
'82'	q the second large prime number such that $q (p-1)$, with q a divisor of (p - 1)
'83'	Z_p^* the set of integers U modulo p such as $0 < U < p$ and $\gcd(U, p) = 1$, $\gcd()$ being the greatest common divisor
'84'	Z_q^* the set of integers U' modulo q such as $0 < U' < q$ and $\gcd(U', q) = 1$
'85'	g the first element of Z_p^* of order q such as g is a generator of G_q and G_q a cyclic group of prime order q
'86'	h the second element of Z_p^* of order q different from g
'87'	y the public key, an integer denoted as $y = g^{-r} h^{-s} \bmod p$ where (s,r) is the secret key, and s and r are two elements of (Z_q^*) , and h of (Z_p^*)
NOTE In this context, ISO/IEC JTC 1/SC 17 reserves any other data object of the context-specific class (first byte in the range '80' to 'BF').	
^a The RSA Okamoto-Schnorr signature scheme, is considered a blind signature process, which is an interactive procedure between a signer and a recipient. It allows a recipient to obtain a signature of a message of the recipient's choice without giving the signer any information about the actual message or the resulting signature [8] [9] [10] [11]. DO'73' may be used in the data field for returning a multi-part digital signature response comprised of concatenation of context-specific data objects defined by the application.	

NOTE 5 For other Blind Signature schemes, e.g. Blind RSA signature (with data objects related to RSA), Blind Schnorr signature (with data objects related to DSA and/or ECDSA), Okamoto-Guillou-Quisquater blind signature scheme (with data objects related to GQ2), the OID under template '7F49' determines the nature and meaning of any further or different data objects, i.e. the following indications are possibly denoted by the OID.

- Blind signature type, e.g. RSA, Schnorr, Okamoto-Schnorr, Okamoto-Guillou-Quisquater.
- Cryptographic Hash function.
- Generic description of the token/credential (message) to be signed.
- Attributes generic structure, and/or
- Type of control upon signed message, i.e. partially blind, fully blind or restrictive blind signature (in some mechanisms, the signer does not totally lose control over the signed message since the signer can include explicit information in the resulting signature based on some agreement with the recipient. Such blind signatures are called partially blind signatures. Other mechanisms allow a recipient to receive a blind signature on a message not known to the signer but the choice of the message is restricted and conforms to certain rules. Such schemes are called restrictive blind signature mechanisms).

A1 For the coding of the DO stating information about the private part of the key pair, Table 4 applies except for QSC. For QSC, the private key shall be nested into template DO'7F75' or DO'7F76' using the DOs as proposed in Table AMD.1.1. **A1**

Table 4 — Private key data objects

Tag	Value
'7F48'	Interindustry template for nesting one set of private key data object with the following tags
'82'	Public exponent (optional)
'92'	Parameter p
'93'	Parameter q
'94'	Parameter $1/q \bmod p$
'95'	Parameter $d \bmod (p-1)$
'96'	Parameter $d \bmod (q-1)$

Table 4 (continued)

Tag	Value
'7F48'	Interindustry template for nesting one set of ECDSA/ECDH private key data object with the following tags
'92'	Private key
'06'	Object identifier of related curve (optional)
	Curve information (conditional if DO'06' above is not specified)
'93'	p is the prime specifying the base field
'94'	A 1st coefficient of the equation $y^2 = x^3 + A*x + B \bmod p$ defining the elliptic curve
'95'	B 2nd coefficient of the equation $y^2 = x^3 + A*x + B \bmod p$
'96'	G = (x,y) base point, i.e. a point in E of prime order, with x and y being its x- and y-coordinates
'97'	q prime order of the group generated by G
'98'	h cofactor of G in E, i.e. $\#E[GF(p)]/q$
NOTE In this context, ISO/IEC JTC 1/SC 17 reserves any other data object of the context-specific class (first byte in the range '80' to 'BF').	

[Annex C](#) provides examples of exporting a public key and importing a private key.

A1 5.2.2 QSC template data objects

This QSC template is a generic template catering to any type of QSC, e.g. hash-based or lattice-based, and its parameters, features general purposes data objects in its key information part whereas the key specific parameters, e.g. for private or public key or common parameters, are nested in template key value part. QSC template data objects are described in Table AMD.1.1. **A1**

Table AMD.1.1 — QSC key components within template DO'7F75', DO'7F76' or DO'7F77' with cardinality indication

Tag	Description	Cardinality	
		at creation	at update
'06'	OID , conditional (present at creation and if DO'80' is absent) ^c	0 or 1	0
'80'	AlgID , conditional (present at creation and if first DO'06' is absent)	0 or 1	0
'81'	Key type , conditional (present at creation, absent at update); unique per template	1 or 2	0
'82'	Key size , conditional (present at creation, absent at update) ^b	From 0 up to n	0
'8E'	Identifier of external common parameters , conditional (optional at creation, absent at update)	0 or 1	0
'8F'	Identifier of the QSC template , conditional (optional at creation and at update)	0 or 1	0 or 1
'5C'	Parameters Organization Descriptor , optional (may nest tag '81', tag '82' or tag '83') ^d '81' denotes private key parameters '82' denotes public key parameters '83' denotes common parameters	From 0 up to n	From 0 up to n
'06'	Application-specific QSC parameters mapping	0 or 1	0 or 1
'9X'	Key value (either private, public key, or common parameters), inferred by Key type (DO'81'), by template tag (DO'7F75', DO'7F76', DO'7F77') and by Parameters Organization Descriptor (DO'5C') ^a	0 or 1	0 or 1

NOTE 1 Use of DO'83' to DO'8D' is RFU, can serve for future indication of further characteristics, e.g. type of memory (transient, persistent);

NOTE 2 If common parameters are merged within private or public key template, the Key Type (DO'81') can be replicated (e.g. hash-based protocol).

NOTE 3 DO'8F' is the Private/Public Key identifier to be used by the external world to request a digital signature computation/verification or encryption/decryption from the ICC, e.g. the value of DO'8F' can be assigned to DO'84' to identify a private key from within a CRT (control reference template). DO'8F' can be assigned to DO'83' value in order to identify the public key from within a CRT.

NOTE 4 Additional private tags that can be used in template '7F75' to nest application specific data, e.g. pre-generated private keys or proprietary parameter(s) involved in key generation or state maintenance, are outside interoperability scope.

NOTE 5 DO'8E' is no longer useful if common parameters and private/public key are nested in the same template.

^a Key value component ('9X') may be provisioned entirely at once or in several parts at creation and update.

^b DO'82' may be replicated for each level of the Merkle Tree with HSS hash-based scheme (see Figures F.6 and F.7).

^c This DO'06' describing the algorithm shall be the first DO in the template; only one OID describing the algorithm shall be present. Furthermore, OID describing the algorithm and DO AlgID are mutually exclusive. When occurring right after a Tag List DO'5C', a DO'06' is meant for QSC parameter mapping and is not mutually exclusive with AlgID; this DO'06' value is out of scope of this document.

^d The usage of DO'81', DO'82' and DO'83' are described under EXAMPLES 4 in the text below.

For hash-based LMS private key under template DO'7F75', see Table AMD.1.8; for hash-based LMS merged private key and common parameters under template DO'7F76', see Table AMD.1.9. For lattice-based Crystals-Dilithium private key under template DO'7F75', see Table AMD.1.10; for lattice-based Crystals-Dilithium private key and common parameters merged under template DO'7F76', see Table AMD.1.12. For lattice-based Falcon private key under template DO'7F75', see Table AMD.1.11; for lattice-based Falcon private key and common parameters merged under template DO'7F76', see Table AMD.1.13.

The templates DO'7F77' nesting only common parameters for hash-based LMS key, lattice-based Crystals-Dilithium, lattice-based Falcon are presented respectively on Table AMD.1.14, Table AMD.1.15 and Table AMD.1.16. ^{A1}

Each QSC template comprises key information data objects and key value data objects with Parameters Organization Descriptor data object. Key information data objects consist of OID, AlgID, Key type, Key size, identifier of external common parameters and identifier of the QSC template data object. Key value data objects are set of DO'9X' inferred by Key type (DO'81'), see Table AMD.1.22 for details. Five different structures can be nested under QSC templates DO'7F75', DO'7F76' or DO'7F77' as follows:

- private key under QSC template DO'7F75';
- public key under QSC template DO'7F75';
- common parameters under QSC template DO'7F77';
- common parameters and private key under QSC template DO'7F76';
- common parameters and public key under QSC template DO'7F76'.

Usually, common parameter(s) are implicitly known as soon as the cryptographic algorithm is known. Therefore template '7F75' may suffice instead of '7F76'.

The problem of scarcity of context-specific tags arises whenever more than 15 parameters are needed to describe a structure within a QSC template, i.e. with lattice-based Crystals-Dilithium. Therefore, to allow for nesting replicated and ordered primitive tags (bit b6 set to 0) while covering more than 15 parameters the optional Tag List data object DO'5C' (see ISO/IEC 7816-4:2020, 8.4.3) serving as Parameters Organization Descriptor shall be used right after the key information to indicate recursive tag numbering (i.e. cyclic tag numbering) whenever employed within the template. The use of DO'5C' may denote rearrangement of the QSC templates that are mapped explicitly in this document or in ISO/IEC 7816-6 (e.g. it can be useful when some mapped parameters are hidden in a QSC template or when common parameter and public or private key parameters are swapped).

{'5C'-L-(tag1- tag2-...- tagN)} denotes replicated tags in the template as follows: tag numbering starts with tag'90' till tag1; then tag numbering resumes with tag'90' right after tag1 and incremented by one till tag2; then tag numbering resumes with tag'90' right after tag2 and incremented by one till next tag in the Tag List DO'5C' etc. See example of implementation of DO'5C' below.

EXAMPLES 1

{'5C'-'01'-('95')} means tags numbering of a sequence starting from DO'90' is incremented by one and replicated once right after tag '95' resulting in a template as follows:

{'7F75'-L-{key information-{'5C'-01-('95')}-{ '90'-L-V}-{ '91'-L-V}-{ '92'-L-V}-{ '93'-L-V}-{ '94'-L-V}-{ '95'-L-V}-{ '90'-L-V}-{ '91'-L-V}-{ '92'-L-V}...}}

{'5C'-'02'-('95' '93')} means tags numbering of a sequence starting from DO'90' is incremented by one and replicated twice right after tag '95' then tag '93' resulting in a template as follows:

{'7F75'-L-{key information-{'5C'-'02'-('95', '93')}-{ '90'-L-V}-{ '91'-L-V}-{ '92'-L-V}-{ '93'-L-V}-{ '94'-L-V}-{ '95'-L-V}-{ '90'-L-V}-{ '91'-L-V}-{ '92'-L-V}-{ '93'-L-V}-{ '90'-L-V}-{ '91'-L-V}-{ '92'-L-V}-{ '93'-L-V}-{ '94'-L-V}...}}

DO'5C' may be used even when less than 15 parameters are present in the template to indicate the way such template shall be updated or expanded with further data objects when replication is deemed useful.

For tag numbering that restarts with a tag different from DO'90', data object '5C' shall use the following coding with the context-specific DO'9F2X' as indicator.

{'5C'-L-(tag1-'9F2X₁'-tag2-'9F2X₂'-...-tagN-'9F2X_n') } denotes replicated tags in the template as follows: tag numbering starts with tag'90' till tag1; then tag numbering resumes with tag'9X₁' right after tag1 and incremented by one till tag2; then tag numbering resumes with tag'9X₂' right after tag2 and incremented by one till next tag in the Tag List DO'5C' etc. See example of implementation of DO'5C' below. 

EXAMPLES 2

^{A1} {5C'-03'-{95'-9F22'}} means tags numbering of a sequence starting from DO'92' is incremented by one and replicated once right after tag '95' resulting in a template as follows:

{7F75'-L-{key information-{5C'-03'-{95'-9F22'}}-{90'-L-V}-{91'-L-V}-{92'-L-V}-{93'-L-V}-{94'-L-V}-{95'-L-V}-{92'-L-V}-{93'-L-V}-{94'-L-V}...}}

{5C'-07'-{95'-9F23'-94'-9F26'-97'}} means:

- for '95'-9F23': right after tag '95', a new sequence is started with tag numbering from '93' (instead of DO'90'), and incremented by one;
- for '94'-9F26': right after tag '94', a new sequence is started with tag numbering from '96' (instead of DO'90'), and incremented by one;
- for '97': right after tag '97', a new sequence is started with tag numbering from '90', and incremented by one.

{7F75'-L-{key information-{5C'-07'-{95'-9F23'-94'-9F26'-97'}}-{90'-L-V}-{91'-L-V}-{92'-L-V}-{93'-L-V}-{94'-L-V}-{95'-L-V}-{93'-L-V}-{94'-L-V}-{96'-L-V}-{97'-L-V}-{90'-L-V}-{91'-L-V}-{92'-L-V}-{93'-L-V}-{94'-L-V}-{95'-L-V}-{96'-L-V}-{97'-L-V}-{98'-L-V}-{99'-L-V}...}}

Where there is no tag before the DO'9F2X' in the DO'5C', i.e. {5C'-L-{9F2X'-...}}, it denotes that tag numbering starts with tag'9X'. See example of implementation of DO'5C' below.

EXAMPLES 3

{5C'-02'-{9F23'}} means that following current DO'5C', tag numbering starts with tag '93', resulting in a template as follows:

{7F75'-L-{key information-{5C'-02'-{9F23'}}-{93'-L-V}-{94'-L-V}-{95'-L-V}...}}

In addition to its properties described above, DO'5C' may serve to organize parameters into dedicated containers within QSC templates, whereby simplifying explicitly template parsing without ambiguity. To this aim, the three data objects DO'81', DO'82' and DO'83' are nested in DO'5C' to indicate respectively the presence of private key parameters, public key parameters and common parameters followed by a series of tags as described above, see examples below.

EXAMPLES 4

{5C'-02'-{81'-92'}} denotes that following DO'5C', the current QSC template contains private key parameters numbered as a sequence starting from DO'90' and incremented by one, and replicated once right after tag '92', resulting in a template as follows:

{7F75'-L-{key information-{5C'-02'-{81', 92'}}-{90'-L-V}-{91'-L-V}-{92'-L-V}-{90'-L-V}-{91'-L-V}-{92'-L-V}-{93'-L-V}-{94'-L-V}...}}

{5C'-01'-{83'}} denotes that following DO'5C', the current QSC template contains, common parameters numbered as a sequence starting from DO'90', resulting in a template as follows:

{7F77'-L-{key information-{5C'-01'-{83'}}-{90'-L-V}-{91'-L-V}-{92'-L-V}-{90'-L-V}-{94'-L-V}...}}

The main benefit of such a tag organization with DO'5C' is that it makes different arbitrary parameters numbering stay interoperable with each other within the same data object generation.

Table AMD.1.2 to Table AMD.1.16 describe key template structures featuring '5C'DO usage.

When using Tag List DO'5C' for parameters ordering into containers, the actual mapping of such parameters (i.e. assignment of a given DO'9X' to a specific QSC component), may depend on the implementation, in which case, the Tag List may be immediately followed with a DO'06' denoting the implementation authority (e.g. a specification) that determines the mapping. Otherwise, the by-default mapping is the one described in this document, for instance Crystals-Dilithium, Falcon, LMS. Crystals-Dilithium key sizes are given as examples in key templates; actual values may be checked against dilithium v3.1. See Reference [16]. ^{A1}

Table AMD.1.2 — QSC template DO'7F75' encapsulating Hash-based LMS public key

Tag	L	Value			Note
0x7F75	Var.				QSC template encapsulating hash-based LMS public key
		Tag	Length	Value	
		0x80	0x04	0xFE0101XX	Algorithm identifier: Hash-based LMS and 'XX' indicating hash function (see 5.2.3)
		0x81	0x02	0x0012	Key type: ALG_TYPE_LMS_PUBLIC (see 5.2.4)
		0x82	0x02	0x0100	Key size (e.g. SHA-256)
		0x8E	Var.		Identifier of external common parameters
		0x8F	Var.		Identifier of the QSC template (same as private key)
		0x5C	0x01	0x82	Parameters Organization Descriptor see Table AMD.1.1
		0x90	Var.		Public key (root of tree)

NOTE 1 This table describes an example of LMS parameters for public key with leaves (level 2) and root (level 1) as per example from Reference [23] TestCase 2 (p.54). As a general description, in a hash-based hierarchical signature system (HSS) comprising a hierarchy of Merkle trees, L represents the number of stages of Merkle trees. The number of leaves of each Merkle tree evaluates to 2^h where h is the height of the Merkle tree, and the leaves represent tuples of signature key (secret and public) on top of those trees. The parameter q is a 32-bit integer (called index) that indicates the leaf of the Merkle tree where the OTS public key can be looked up. Each leaf of the tree is comprised of the value of the public key of an LM-OTS public/private key pair. Each node within the tree has a unique node number, and the leaves have node numbers 2^h , $(2^h)+1$, $(2^h)+2$, ..., $(2^h)+(2^h)-1$. In general, the j-th node at level i has node number $2^i + j$. Thus the root node has node number 1 (i.e. "root level = $2^0 = 1$ "), see Reference [23] for details.

NOTE 2 0xABCD denotes the hexadecimal number 'ABCD' for easier reference from implementers.

See hash-based schemes taxonomy in Annex F, Figure F.1.

Table AMD.1.3 — QSC template DO'7F75' encapsulating Lattice-based Crystals-Dilithium public key

Tag	L	Value			Note
0x7F75	Var.				QSC template encapsulating Lattice-based Crystals-Dilithium public key
		Tag	Length	Value	
		0x80	0x04	see 5.2.3	AlgID
		0x81	0x02	see Table AMD.1.22	keytype "Crystals-Dilithium public key"
		0x82	0x02	0x04A0	keysize ($1312 = \rho + t_1$) values: 1312, 1952, 2592
		0x8E	0x01	0x02	Identifier of external common parameters
		0x8F	0x01	0x01	Identifier of the QSC template (same as private = keypair)
		0x5C	0x01	0x82	see Table AMD.1.1
		0x90	0x20	0x.....	$\rho(\text{length}=32)$, same for all Crystals-Dilithium NIST levels
		0x91	0x820500	0x.....	$t_1(\text{length}=1280)$ for NIST levels 2, 3 and 5, values: 1280, 1920, 2560

NOTE 1 See keysize according to NIST security levels in Reference [16].

NOTE 2 0xABCD denotes the hexadecimal number 'ABCD' for easier reference from implementers.

Ⓐ

Table AMD.1.4 — QSC template DO'7F75' encapsulating Lattice-based Falcon public key

Tag	L	Value			Note
0x7F75	Var.	Tag	Length	Value	
		0x80	0x04	0xFE0105XX	AlgID Lattice-based FALCON and ‘XX’ indicating hash function (see 5.2.3)
		0x81	0x02	see Table AMD.1.22	keytype "Falcon public key"
		0x82	0x02	0x0381	keysize (=897) values = 897,1793
		0x8E	0x01	0x02	Identifier of external common parameters
		0x8F	0x01	0x01	Identifier of the QSC template (same as private = keypair)
		0x5C	0x01	0x82	
		0x90	0x820381	0x.....	$h(\text{length}=897)$ value = 897,1793
NOTE 1 For recommended Falcon parameters see, e.g. public keysize on ^[17] .					
NOTE 2 0xABCD denotes the hexadecimal number ‘ABCD’ for easier reference from implementers.					

Table AMD.1.5 — QSC template DO'7F76' encapsulating Hash-based LMS public key and common parameters

Tag	L	Value			Note
0x7F76	Var.	Tag	Length	Value	
		0x80	0x04	see 5.2.3	AlgID
		0x81	0x02	see Table AMD.1.22	keytype "LMS public key"
		0x82	0x01	0x20	keysize (e.g. SHA256)
		0x8F	0x01	0x01	Identifier of the QSC template (same as private = keypair)
		0x5C ^a	0x01	0x83	Parameters Organization Descriptor; indicates common parameters, see Table AMD.1.1
		0x90	0x10	0xD08F...	<i>LMS Identifier level1</i>
		0x91	0x04	0x00000006	<i>LMS type level 1 (e.g. LM_SHA256_M32_H10)</i>
		0x92	0x04	0x00000003	<i>LM-OTS type level 1 (e.g. LMOTS_SHA256_N32_W4)</i>
		0x5C	0x01	0x82	Parameters Organization Descriptor; indicates public key parameters, see Table AMD.1.1
		0x90	0x20	0x32A5...	<i>public key (root of the tree)</i>
		0x5C	0x01	0x83	Parameters Organization Descriptor; indicates common parameters, see Table AMD.1.1
		0x90	0x10	0x215F...	<i>LMS Identifier level2</i>
		0x91	0x04	0x00000005	<i>LMS type level 2</i>
		0x92	0x04	0x00000004	<i>LM-OTS type level 2</i>
NOTE 1 Parameter values borrowed from example in Reference [23] TestCase 2 (p.54) with leaves = level 2 and root = level 1, see Note 1 to Table AMD.1.2.					
NOTE 2 DO'8E' (identifier of external common parameters) is no longer useful if common parameters and public key are nested in the same template.					
NOTE 3 0xABCD denotes the hexadecimal number 'ABCD' for easier reference from implementers.					
^a Instead of using multiple instances of Tag List DO'5C', one unique instance may be coded as '5C058392829083' for the same purpose.					

Table AMD.1.6 — QSC template DO'7F76' Lattice-based Crystals-Dilithium public key and common parameters

Tag	L	Value			Note
0x7F76	Var.	Tag	Length	Value	
		0x80	0x04	see 5.2.3	AlgID
		0x81	0x02	see Table AMD.1.22	keytype "Crystals-Dilithium Public key"
		0x82	0x02	0x0531	keysize (e.g. 1312+17 = rho + t1) keysize may take in account size public + size parameters = sum (size of all 0x9X data elements)
		0x8F	0x01	0x01	Identifier of the QSC template
		0x5C	0x01	0x83	Common parameters ^a
		0x90	0x03	0x7FE001	$q (=8380417)$
		0x91	0x01	0x0E	$d(=14)$
		0x92	0x01	0x3C	$weight\ of\ c (=60)$
		0x93	0x03	0x07FE00	$g1(=523776)$
		0x94	0x03	0x03FF00	$g2(=261888)$
		0x95	0x01	0x04	$k(=4)$
		0x96	0x01	0x03	$l(=3)$
		0x97	0x01	0x06	$h(=6)$
		0x98	0x02	0x0145	$b(=325)$
		0x99	0x01	0x50	$w(=80)$
		0x5C	0x01	0x82	Public key parameters
		0x90	0x20	0x.....	$\rho(length=32)$
		0x91	0x820500	0x.....	$t1(length=1280)$

NOTE 1 See Reference [16].

NOTE 2 0xABCD denotes the hexadecimal number 'ABCD' for easier reference from implementers.

^a This template '7F76' is not necessary as the common parameter(s) listed from DO'90' to DO'99' are likely to be implicitly known as soon as the algorithm is known. Therefore template '7F75' may suffice.

Table AMD.1.7 — QSC template DO'7F76' encapsulating Lattice-based Falcon public key and common parameters

Tag	L	Value			Note
0x7F76	Var.	Tag	Length	Value	
		0x80	0x04	0xFE0105XX	AlgID Lattice-based FALCON and 'XX' indicating hash function (see 5.2.3)
		0x81	0x02	see Table AMD.1.22	keytype "Falcon public key"
		0x82	0x02	0x0381	keysize (=897)
		0x8F	0x01	0x01	Identifier of the QSC template
		0x5C	0x01	0x83	Common parameters ^a
		0x90	0x02	0x3001	$q (=8380417)$
		0x91	0x02	0x0200	$d(=14)$
		0x92	0x04	0x029845D6	$weight\ of\ c (=60)$
		0x5C	0x01	0x82	

NOTE 1 See Note 1 to Table AMD.1.4.

NOTE 2 0xABCD denotes the hexadecimal number 'ABCD' for easier reference from implementers.

^a This template '7F76' is not necessary as the common parameter(s) listed from DO'90' to DO'92' are likely to be implicitly known as soon as the algorithm is known. Therefore template '7F75' may suffice.

A1

A1 Table AMD.1.7 (continued)

Tag	L	Value			Note
		0x90	0x820381	0x.....	$h(\text{length}=897)$
NOTE 1 See Note 1 to Table AMD.1.4.					
NOTE 2 0xABCD denotes the hexadecimal number 'ABCD' for easier reference from implementers.					
^a This template '7F76' is not necessary as the common parameter(s) listed from DO'90' to DO'92' are likely to be implicitly known as soon as the algorithm is known. Therefore template '7F75' may suffice.					

Table AMD.1.8 — QSC template DO'7F75' encapsulating Hash-based LMS private key

Tag	L	Value			Note
0x7F75	Var.	Tag	Length	Value	
		0x80	0x04	see 5.2.3	AlgID
		0x81	0x02	see Table AMD.1.22	keytype "LMS private key"
		0x82	0x20		keysize (e.g. SHA256) (may be omitted)
		0x8E	0x01	0x02	Identifier of external common parameters
		0x8F	0x01	0x01	Identifier of the QSC template (same as public = keypair)
		0x5C	0x01	0x81	see Table AMD.1.1
		0x90	0x20	0x558B8...	privatekey level 1 (see Reference [23], Appendix A)
		0x91	0x20	0xA1C4...	privatekey level 2 (see Reference [23], Appendix A)
NOTE 1 With leaves = level 2 and root = level 1, see Note 1 to Table AMD.1.2.					
NOTE 2 0xABCD denotes the hexadecimal number ‘ABCD’ for easier reference from implementers.					

For hash-based scheme example of key generation, see F.2.

Table AMD.1.9 — QSC template DO'7F76' encapsulating Hash-based LMS merged private key and common parameters

Tag	L	Value			Note
0x7F76	Var.	Tag	Length	Value	
		0x80	0x04	see 5.2.3	AlgID
		0x81	0x02	see Table AMD.1.22	keytype "LMS private key"
		0x82	0x01	0x20	keysize (e.g. SHA256) (may be omitted)
		0x8F	0x01	0x01	Identifier of the QSC template
		0x5C	0x01	0x83	see Table AMD.1.1
		0x90	0x10	0xD08F...	<i>LMS Identifier level1</i>
		0x91	0x04	0x00000006	<i>LM type level 1 (e.g. LM_SHA256_M32_H10)</i>
		0x92	0x04	0x00000003	<i>LM-OTS type level 1 (e.g. LMOTS_SHA256_N32_W4)</i>
		0x5C	0x01	0x81	see Table AMD.1.1
		0x90	0x20	0x558B8...	<i>privatekey level 1</i> (see Reference [23], Appendix A)
		0x5C	0x01	0x83	see Table AMD.1.1
		0x90	0x10	0x215F...	<i>LMS Identifier level2</i>
		0x91	0x04	0x00000005	<i>LMS type level 2</i>
		0x92	0x04	0x00000004	<i>LM-OTS type level 2</i>
		0x5C	0x01	0x81	see Table AMD.1.1
		0x90	0x20	0xA1C4...	<i>privatekey level 2</i> (see Reference [23], Appendix A)
NOTE 1 See Note 1 to Table AMD.1.2.					
NOTE 2 0xABCD denotes the hexadecimal number ‘ABCD’ for easier reference from implementers.					

Table AMD.1.10 — QSC template DO'7F75' encapsulating Lattice-based Crystals-Dilithium private key

Tag	L	Value			Note
0x7F75	Var.	Tag	Length	Value	
		0x80	0x04	see 5.2.3	AlgID
		0x81	0x02	see Table AMD.1.22	keytype "Crystals-Dilithium private key"
		0x82	0x02	0x09BC	keysize (=2492)
		0x8E	0x01	0x02	Identifier of external common parameters
		0x8F	0x01	0x01	Identifier of the QSC template (same as public = keypair)
		0x5C	0x01	0x81	see Table AMD.1.1
		0x90	0x20	0x.....	$\rho(\text{length}=32)$ same for all NIST levels
		0x91	0x20	0x.....	$K(\text{length}=32)$ same for all NIST levels
		0x92	0x20	0x.....	$tr(\text{length}=32)$ same for all NIST levels
		0x93	0x820180	0x.....	$s1(\text{length}=3*128=384)$
		0x94	0x820180	0x.....	$s2(\text{length}=3*128=384)$
		0x95	0x820680	0x.....	$t0(\text{length}=1664)$ values = 1344,1792,2240,2688
NOTE 1 See Reference [16].					
NOTE 2 0xABCD denotes the hexadecimal number ‘ABCD’ for easier reference from implementers.					

Table AMD.1.11 — QSC template DO'7F75' encapsulating Lattice-based Falcon private key

Tag	L	Value			Note
0x7F75	Var.	Tag	Length	Value	
		0x81	0x02	see Table AMD.1.22	keytype "Falcon private key"
		0x82	0x02	0x1001	keysize (=4097) values = 4097,8193
		0x8E	0x01	0x02	Identifier of external common parameters
		0x8F	0x01	0x01	Identifier of the QSC template (same as public = keypair)
		0x5C	0x01	0x81	see Table AMD.1.1
		0x90	0x820200	0x.....	$f(\text{length}=512)$ values = 512,1024
		0x91	0x820200	0x.....	$g(\text{length}=512)$ values = 512,1024
		0x92	0x820200	0x.....	$F(\text{length}=512)$ values = 512,1024
		0x93	0x820200	0x.....	$G(\text{length}=512)$ values = 512,1024
NOTE 0xABCD denotes the hexadecimal number ‘ABCD’ for easier reference from implementers.					

Table AMD.1.12 — QSC template DO'7F76' encapsulating Lattice-based Crystals-Dilithium private key and common parameters

Tag	L	Value			Note
0x7F76	Var.	Tag	Length	Value	
		0x80	0x04	see 5.2.3	AlgID optional
		0x81	0x02	see Table AMD.1.22	keytype "Crystals-Dilithium private key"
		0x82	0x02	0x09CD	keysize (2492+17)
		0x8F	0x01	0x01	Identifier of the QSC template
		0x5C	0x01	0x83	see Table AMD.1.1 ^a
NOTE 0xABCD denotes the hexadecimal number ‘ABCD’ for easier reference from implementers.					
^a This template ‘7F76’ is not necessary as the common parameter(s) listed from DO’90’ to DO’99’ are likely to be implicitly known as soon as the algorithm is known. Therefore template ‘7F75’ may suffice.					

A1 Table AMD.1.12 (continued)

Tag	L	Value			Note
		0x90	0x03	0x7FE001	$q (=8380417)$
		0x91	0x01	0x0E	$d (=14)$
		0x92	0x01	0x3C	weight of $c (=60)$
		0x93	0x03	0x07FE00	$g1 (=523776)$
		0x94	0x03	0x03FF00	$g2 (=261888)$
		0x95	0x01	0x04	$k (=4)$
		0x96	0x01	0x03	$l (=3)$
		0x97	0x01	0x06	$h (=6)$
		0x98	0x02	0x0145	$b (=325)$
		0x99	0x01	0x50	$w (=80)$
		0x5C	0x01	0x81	see Table AMD.1.1
		0x90	0x20	0x.....	$\rho(\text{length}=32)$
		0x91	0x20	0x.....	$K(\text{length}=32)$
		0x92	0x20	0x.....	$tr(\text{length}=32)$
		0x93	0x820180	0x.....	$s1(\text{length}=3*128=384)$
		0x94	0x820180	0x.....	$s2(\text{length}=3*128=384)$
		0x95	0x820680	0x.....	$t0(\text{length}=1664)$

NOTE 0xABCD denotes the hexadecimal number 'ABCD' for easier reference from implementers.

^a This template '7F76' is not necessary as the common parameter(s) listed from DO'90' to DO'99' are likely to be implicitly known as soon as the algorithm is known. Therefore template '7F75' may suffice.

Table AMD.1.13 — QSC template DO'7F76' encapsulating Lattice-based Falcon private key and common parameters

Tag	L	Value			Note
0x7F76	Var.	Tag	Length	Value	
		0x80	0x04	0xFE0105XX	AlgID Lattice-based FALCON and 'XX' indicating hash function (see 5.2.3)
		0x81	0x02	see Table AMD.1.22	keytype "Falcon private key"
		0x82	0x02	0x1001	keysize (=4097)
		0x8F	0x01	0x01	Identifier of the QSC template
		0x5C	0x01	0x83	see Table AMD.1.1 ^a
		0x90	0x02	0x3001	$q (=12289)$
		0x91	0x02	0x0200	$n (=512)$
		0x92	0x04	0x029845D6	bound $b (=43533782)$
		0x5C	0x01	0x81	see Table AMD.1.1
		0x90	0x820200	0x.....	$f(\text{length}=512)$
		0x91	0x820200	0x.....	$g(\text{length}=512)$
		0x92	0x820200	0x.....	$F(\text{length}=512)$
		0x93	0x820200	0x.....	$G(\text{length}=512)$

NOTE 0xABCD denotes the hexadecimal number 'ABCD' for easier reference from implementers.

^a This template '7F76' is not necessary as the common parameter(s) listed from DO'90' to DO'92' are likely to be implicitly known as soon as the algorithm is known. Therefore template '7F75' may suffice.

A1