



DPC: 24/30506654 DC

BSI Group Headquarters

389 Chiswick High Road London W4 4AL

Tel: +44 (0)20 8996 9000

Fax: +44 (0)20 8996 7400

www.bsigroup.com

Date: 09 December 2024

Origin: International

Latest date for receipt of comments: 04 January 2025

Project No. 2023/00705

Responsible committee: IST/12 Financial services

Interested committees:

Title: Draft BS ISO 20038 Banking and related financial services — Key wrap using advanced encryption standard (AES).

Please notify admin.start@bsigroup.com if you are aware of any keywords that might assist in clarifying or identifying the standard or of the content of this standard:

- i) has any issues related to 3rd party IPR, patent or copyright
- ii) affects other national standard(s)
- iii) requires additional national guidance or information

WARNING: THIS IS A DRAFT AND MUST NOT BE REGARDED OR USED AS A BRITISH STANDARD. THIS DRAFT IS NOT CURRENT BEYOND 04 January 2025

This draft is issued to allow comments from interested parties; all comments will be given consideration prior to publication. No acknowledgement will normally be sent. **See overleaf for information on the submission of comments.**

No copying is allowed, in any form, without prior written permission from BSI except as permitted under the Copyright, Designs and Patent Act 1988 or for circulation within a nominating organization for briefing purposes. Electronic circulation is limited to dissemination by e-mail within such an organization by committee members.

Further copies of this draft may be purchased from BSI Shop <http://shop.bsigroup.com> or from BSI Customer Services, Tel: +44(0) 20 8996 9001 or email cservices@bsigroup.com. British, International and foreign standards are also available from BSI Customer Services.

Information on the co-operating organizations represented on the committees referenced above may be obtained from <https://standardsdevelopment.bsigroup.com/>.

E-mail: admin.start@bsigroup.com

Introduction

This draft standard is based on national and international discussions. Your comments on this draft are invited and will assist in the preparation of the consequent standard.

For international standards, comments will be reviewed by the relevant UK national committee before sending the consensus UK vote and comments to the international committee, which will then decide appropriate action. If the international standard is approved, it is usual for the text to be published as a British Standard.

For national standards, comments will be reviewed by the relevant UK national committee and the resulting standards published as a British Standard.

UK Vote

Please indicate whether you consider the UK should submit a negative (with supporting technical reasons) or positive vote on this draft. Please indicate if you are aware of any reason why this draft standard should not be published as a British Standard.

Submission of Comments

- **Annotated drafts are not acceptable and will be rejected.**
- All comments should be submitted online at <https://standardsdevelopment.bsigroup.com/>. You will need to register in order to comment.

Template for comments and secretariat observations

Date: xx/xx/20xx	Document: ISO/DIS xxxx
------------------	------------------------

1	2	(3)	4	5	(6)	7
MB	Clause No./ Subclause No./Annex (e.g. 3.1)	Paragraph/ Figure/Table/Note	Type of comment	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
	3.1	Definition 1	ed	Definition is ambiguous and needs clarifying.	Amend to read '...so that the mains connector to which no connection...'	
	6.4	Paragraph 2	te	The use of the UV photometer as an alternative cannot be supported as serious problems have been encountered in its use in the UK.	Delete reference to UV photometer.	



DRAFT International Standard

ISO/DIS 20038.2

Banking and related financial services — Key wrap using advanced encryption standard (AES)

ICS: 35.240.40

ISO/TC 68/SC 2

Secretariat: **BSI**

Voting begins on:
2024-12-09

Voting terminates on:
2025-02-03

This document is circulated as received from the committee secretariat.

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENTS AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

Reference number
ISO/DIS 20038.2:2024(en)



COPYRIGHT PROTECTED DOCUMENT

© ISO 2024

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	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms, definitions and abbreviations	1
3.1 Terms and definitions	1
3.2 Abbreviations	5
4 Notation	5
5 Key Block Elements	6
6 Key Block Format	7
6.1 Introduction	7
6.2 Key block header (KBH)	8
6.2.1 General	8
6.2.2 Changing Key Headers	11
6.3 Defined values for key block headers	12
6.3.1 Key Usage	12
6.3.2 Algorithm	17
6.3.3 Mode of use	17
6.3.4 Key Version Number	19
6.3.5 Exportability	19
6.3.6 Optional block ID	20
6.4 Encoding	43
6.4.1 General	43
6.4.2 RSA Key Pairs	44
6.4.3 Elliptic Curve Cryptography (ECC) Key Pairs	45
7 Key Block Binding and Validation Methods	45
7.1 General	45
7.2 Key Block Binding Method Using Key Derivation	46
7.2.1 General	46
7.2.2 Key Derivation Binding Method – AES	46
7.2.3 Key Derivation Binding Validation Method - AES	50
8 Examples	51
8.1 AES Key Block Example	51
8.1.1 Introduction	51
8.1.2 Constructing the key block header	51
8.2 AES key block with optional blocks	57
8.2.1 Introduction	57
8.2.2 Keys used in the example	57
8.2.3 Constructing the key block header	58
8.3 RSA key block Example with CT optional block	60
8.3.1 General	60
8.3.2 Keys used in the example	60
8.3.3 Constructing the key block header	61
8.3.4 Constructing the binary key data	63
8.3.5 Constructing the complete key block	64
8.4 ECC key block Example with CT certificate chain optional block	66
8.4.1 General	66
8.4.2 Keys used in the example	66
8.4.3 Constructing the key block header	67
8.4.4 Constructing the binary key data	68
8.4.5 Constructing the complete key block	69
8.5 CTR mode without padding	70

Annex A (normative) Directed Key Diversification	71
Annex B (informative) Operational challenges and examples with exportability	79
Bibliography	88

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.

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 ISO 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).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

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.

This document was prepared by Technical Committee ISO/TC 68, *Financial services*, Subcommittee SC 2, *Financial services, security*.

This second edition cancels and replaces the first edition (ISO 20038:2017), which has been technically revised.

The main changes are as follows:

- updated for compatibility with ANSI X9.143-2022.

A list of all parts in the ISO 20038 series can be found on the ISO website.

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.

Introduction

The secure management of cryptographic keys requires that their values and usage constraints be protected for both confidentiality and integrity.

This document provides a method of wrapping cryptographic keys in order to provide confidentiality and integrity protection for the keys when being transmitted or stored. The mechanism is designed to use advanced encryption standard (AES) specified in FIPS 197 as the wrapping cipher.

Banking and related financial services — Key wrap using advanced encryption standard (AES)

1 Scope

This document defines a method for packaging cryptographic keys for transport. This method can also be used for the storage of keys under an advanced encryption standard (AES) key. The method uses the block cipher AES as the wrapping cipher algorithm.

Other methods for wrapping keys are outside the scope of this document but can use the authenticated encryption algorithms specified in ISO/IEC 19772.

2 Normative references

There are no normative references in this document.

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the following terms, definitions and abbreviations 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/>

3.1.1 advanced encryption standard AES

16-byte block cipher

Note 1 to entry: This is defined in ISO/IEC 18033-3

[SOURCE: ISO 11568:2023, 3.2]

3.1.2 ASCII American Standard Code for Information Interchange

binary encoding of 7-bit characters defined by ISO/IEC 646

[SOURCE: ISO/IEC 18477-4:2017(en), 3.1.2]

3.1.3 authentication value

cryptographically generated value (*message authentication code* (3.1.23)) derived from the key block header and confidential data that can be used to validate the integrity of the data contained in the key block

3.1.4

CBC

cipher block chaining

confidentiality mode whose encryption process features the combining (“chaining”) of the plaintext blocks with the previous ciphertext blocks

[SOURCE: NIST SP800-38A: 2001]

3.1.5

cryptographic domain

set of one or more *secure cryptographic devices* ([3.1.26](#)) that contain or are able to use the same set of keys

Note 1 to entry: All *secure cryptographic devices* ([3.1.26](#)) in the domain are considered to be “equivalent” in that a key that can be used in one *secure cryptographic device* ([3.1.26](#)) in the domain is also available for use by any other *secure cryptographic device* ([3.1.26](#)) in the domain.

Note 2 to entry: A cryptographic domain is often defined for host processing to provide load balancing of HSMs.

Note 3 to entry: The keys are typically structured in a hierarchy where higher-level keys are used to provide protection for lower-level keys.

3.1.6

CTR

counter mode of encryption

confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa

[SOURCE: NIST SP800-38A:2001]

3.1.7

DEA

Data Encryption Algorithm

Cryptographic algorithm originally specified in FIPS 46, *The Data Encryption Standard*, which became effective July 1977

[SOURCE: NIST SP 800-67 Rev. 2]

3.1.8

hex-ASCII

number in base 16 expressed as ASCII ([3.2](#)) characters

EXAMPLE The 32-bit *message authentication code* ([3.1.23](#)) result 0x12345678 (4 binary bytes) is encoded as eight hex-ASCII characters that reads "12345678" but is stored or transmitted as 0x3132333435363738.

3.1.9

initialization vector

IV

binary value used as a starting point for the encryption of a data sequence in order to increase security by introducing additional cryptographic variance and to synchronize cryptographic equipment

3.1.10

key block

result of performing a cryptographic process over a key and its associated data to produce a key block header (KBH), ciphertext and *authentication value* ([3.1.3](#)) – the *message authentication code* ([3.1.23](#))

[SOURCE: ANSI X9.143:2022, 3.15]

3.1.11**key block encryption key****KBEK**

key that is derived from the *key block protection key* ([3.1.13](#)) and that is used solely for enciphering the *key block* ([3.1.10](#))

[SOURCE: ANSI X9.143:2022, 3.16]

3.1.12**key block authentication key****KBAK**

key that is derived from the *key block protection key* ([3.1.13](#)) and that is used solely for calculating the *message authentication code* ([3.1.23](#)) over the *key block* ([3.1.10](#))

[SOURCE: ANSI X9.143:2022, 3.17]

3.1.13**key block protection key****KBPk**

key wrapping key ([3.1.20](#)) that is used as a derivation key from which the *key block encryption key* ([3.1.11](#)) and the *key block authentication key* ([3.1.12](#)) are derived

Note 1 to entry: This key is used for no other purpose.

[SOURCE: ANSI X9.143:2022, 3.18]

3.1.14**key diversification**

HSM secured operation to derive a key from another key by means of a key derivation function and derivation data that contains the key attributes that shall be assigned to the derived key by the HSM.

3.1.15**key diversification key****KDK**

key used in an HSM function implementing the directed key diversification scheme (see [Annex A](#)) where a key is derived from the key diversification key according to a type A or B, bound to the key diversification key, and a *key type vector* ([3.1.19](#)) that is input to the derivation function and used by the HSM to control the attributes of the derived key

3.1.16**key encrypting key****KEK**

key used exclusively to encrypt and decrypt other keys

[SOURCE: ANSI X9.143:2022, 3.19]

3.1.17**key export**

secure cryptographic device ([3.1.26](#)) operation that a) unwraps a key and its attributes from a storage *key encrypting key* ([3.1.16](#)) or the *master file key* ([3.1.22](#)) and re-encrypts that key under a *key wrapping key* ([3.1.20](#)) shared with another entity or internal system; or b) wraps a clear key and its attributes that is stored within an *secure cryptographic device* ([3.1.26](#)) using a *key wrapping key* ([3.1.20](#)) shared with another entity or internal system

[SOURCE: ANSI X9.143:2022, 3.20]

3.1.18**key import**

secure cryptographic device (3.1.26) operation that unwraps a key and its attributes using a *key wrapping key* (3.1.20) shared with another entity or system and then stores the key and its attributes from the key block within one or more related *secure cryptographic devices* (3.1.26) or re-wraps the key and its attributes using a *storage key encrypting key* (3.1.16) or the *master file key* (3.1.22)

[SOURCE: ANSI X9.143:2022, 3.21]

3.1.19**key type vector****KTV**

Set of attributes formatted according to the directed *key diversification* scheme (see [Annex A](#)) and used as input to the diversification of a *key diversification key* in generating a derived key

3.1.20**key wrapping key**

symmetric key that determines the *wrapping* (3.1.30) and *unwrapping* (3.1.29) functions of a *key wrapping mechanism* (3.1.21)

[SOURCE: ANSI X9.143:2022, 3.22]

3.1.21**key wrapping mechanism**

symmetric key authenticated encryption mechanism that is intended for the protection of cryptographic keys and other specialized data

[SOURCE: ANSI X9.143:2022, 3.23]

3.1.22**master file key****MFK**

key used to encrypt other keys for storage

Note 1 to entry: Storage may be internal to the *secure cryptographic device* (3.1.26) housing the master file key or may pertain to external storage (e.g. on a host system).

[SOURCE: ANSI X9.143:2022, 3.24]

3.1.23**message authentication code****MAC**

cryptographic value which is the result of passing a financial message through the message authentication algorithm using a specific key

[SOURCE: ANSI X9.143:2022, 3.25]

3.1.24**PIN****personal identification number**

numeric code used to authenticate an identity

[SOURCE: ISO/IEC 19790:2012]

3.1.25**pseudorandom function**

function that can be used to generate output from a random seed and a data variable such that the output is computationally indistinguishable from truly random output

3.1.26**secure cryptographic device****SCD**

device that provides physically and logically protected cryptographic services and storage [e.g. *PIN* ([3.1.24](#)) entry device (PED) or hardware security module (HSM)], and which can be integrated into a larger system, such as an automated teller machine (ATM) or point of sale (POS) terminal

[SOURCE: ISO 13491-1:2016, 3.28]

3.1.27**subkey**

cryptographic key that is used internally in a cryptographic algorithm, such as CMAC, and not chosen by or exposed to the user of the algorithm

3.1.28**TDEA****Triple Data Encryption Algorithm**

symmetric algorithm

Note 1 to entry: The TDEA is specified in the ISO/IEC 18033 series

[SOURCE: ISO 11568:2023, 3.87]

3.1.29**unwrapping**

function in a key wrap mechanism that either produces the key data from the ciphertext or indicates that either the ciphertext or the associated data is invalid

[SOURCE: ANSI X9.143:2022, 3.32]

3.1.30**wrapping**

function in a key wrap mechanism that produces the ciphertext from and creates a validity check for the key and associated data

[SOURCE: ANSI X9.143:2022, 3.33]

3.2 Abbreviations

For the purposes of this document, the following abbreviations apply:

ECC	elliptic curve cryptography
EMV	Europay, Mastercard, Visa
HMAC	keyed-hash message authentication code
ID	identifier
KBH	key block header
RSA	Rivest, Shamir and Adelman

4 Notation

The following are used to indicate field encoding in the key block:

- a) nA – n-digits of alphabetic ('A' to 'Z', 'a' to 'z'), for example 6A means exactly six alphabetic characters in ASCII.

- b) nAN – alphanumeric ('A' to 'Z', 'a' to 'z', '0' to '9'), for example 6AN means exactly six alphanumeric characters in ASCII.
- c) nH – hex-ASCII ('0' to '9', 'A' to 'F'), for example 6H means exactly six hex-ASCII characters.
- d) nN – numeric-ASCII ('0'-'9'), for example 6N means exactly six decimal characters in ASCII.
- e) nB – binary bytes (0x00 to 0xFF), for example 6B means exactly six bytes of binary data.
- f) nPA – n printable ASCII characters. Printable ASCII characters are those with hexadecimal values in the range 20 to 7E, inclusive. 6PA means exactly six printable ASCII characters.
- g) || – used for concatenation of bitstrings, for example 30A4F || 0C1 is equal to 30A4F0C1.
- h) $\oplus$ – xor. Bit-wise, xor is defined as $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$ and $1 \oplus 1 = 0$. $\oplus$ applied to longer strings is performed bitwise, for example $0x3FC4 \oplus 0xED27 = 0xD2E3$.
- i) $\ll$ – left-shift. If M is a bit string, then $M \ll 1$ means M shifted one bit to the left. For example, the bit string 0101010 $\ll 1$ is 1010100 and $1010100 \ll 1$ is 0101000. Notice that the length of $M \ll 1$ is the same as the length of M. Left-shift can also be described as multiplying by 2, without carry.
- j) 0x – Notation indicating a hexadecimal number follows. For example, 0x31 indicates 31-hexadecimal (49-decimal).

5 Key Block Elements

The key block shall contain three parts:

- a) the key block header (KBH) which contains non-sensitive attribute information about the key and the key block;
- b) the confidential data that is being exchanged or stored (i.e. the key and attributes formatted according to the key data format);
- c) the authentication value which consists of a MAC that binds the key block header and confidential data.

This general format of a key block is illustrated in [Figure 1](#).

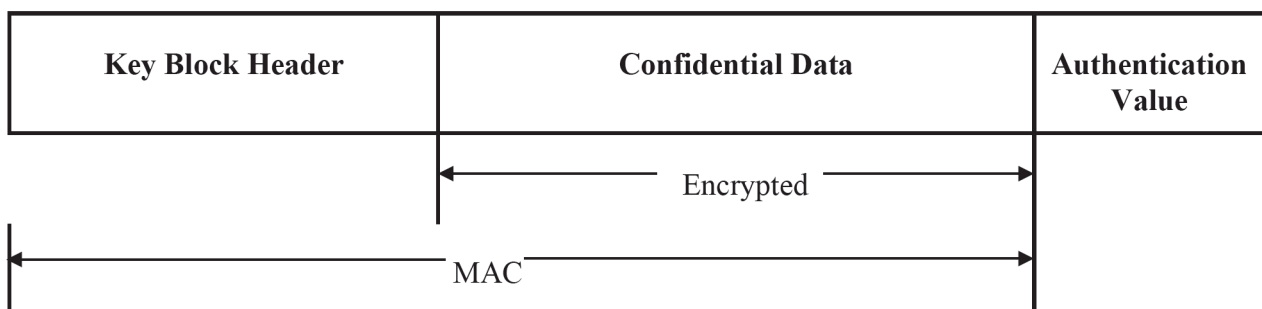


Figure 1 — Key Block

For TDEA and AES keys the confidential data should be padded to the maximum key length to hide its true length (key length obfuscation padding) from an external observer. When the confidential data is padded for key length obfuscation:

- TDEA keys less than 192 bits shall be padded to 192 bits.
- AES keys less than 256 bits shall be padded to 256 bits.
- Requirements for padding values are defined in detailed description in the next paragraph.

The key length obfuscation padding is determined by the size of the key being wrapped, as described in the previous paragraph, but the block padding is determined by the algorithm being used to encrypt the confidential data. Because of the existence of the key length field in the information that is encrypted, additional padding is necessary to make the block consistent with the size of block the algorithm requires. For example, AES operates on 128 bits, so if the key is padded out to 256 bits, because of the key length field additional block padding characters are required.

For operational and compliance reasons, it is permissible for an entity to have SCD mechanisms to:

- a) verify key length, such as against an organization's policies and industry requirements;
- b) obtain key length information of stored keys.

Such SCD mechanisms should be managed with SCD administrative controls requiring dual-control execution.

6 Key Block Format

6.1 Introduction

This clause defines a secure key block that uses the methods described in [clause 7](#).

Key blocks constructed according to this document shall also conform to the relevant key wrapping requirements in ANSI X9.102.

The key block is a variable-length structure with a maximum length of 9999 bytes. The key block is composed of three sub-structures: a variable-length header, a variable-length confidential data field, and an authentication value. The header may include optional fields.

The key block shall consist of three parts:

- a) The key block header (KBH), which contains attribute information about the key and the key block and is not encrypted
 - 1) The first section is 16 bytes with a fixed format defined in [Table 1](#)
- b) The second section is variable length and optional
 - 1) The confidential data which will be encrypted
 - 2) Two bytes indicating the key length
 - 3) The secret key and/or sensitive data
 - 4) Random key length obfuscation padding as required.
 - 5) Random cipher block padding.
- c) The authentication value. The length of the authentication value shall be as follows:
 - 1) 128 bits because the AES key derivation method is used (see [subclause 7.2.2](#))

TDEA and AES symmetric keys should be padded with random key length obfuscation padding to the maximum length for the algorithm, 192 bits for TDEA or 256 bits for AES. Other key types will not be padded for key length obfuscation but are padded up to the cipher block of the wrapping method (e.g. HMAC, RSA and ECC keys). This format is illustrated in [Figure 2](#).

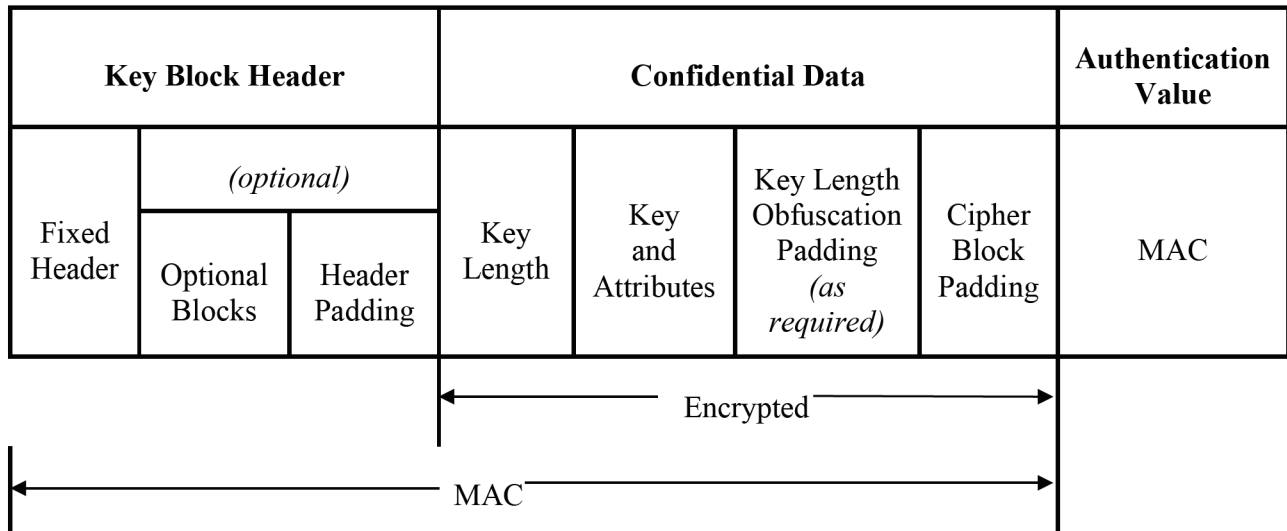


Figure 2 — Key Block

6.2 Key block header (KBH)

6.2.1 General

The header shall contain attribute information about the key as defined in [Table 1](#). The specific encoding and acceptable characters are defined individually for each field, for example in the “Encoding” column of [Table 1](#). When the term “numeric” is used to describe an ASCII header byte, it means an ASCII character in the range “0” to “9” (0x31 to 0x39). A multi-byte header field is considered numeric only when all bytes of the field contain numeric values. [Table 1](#) shows the format of the KBH. See [subclause 6.3](#) for defined key attributes.

Table 1 — Key block header

Byte #	Field name	Description	Encoding	Encrypted
0	Key block version ID (identifier)	Identifies the version of the key block, which defines the method by which it is cryptographically protected and layout of the block: ‘D’ (0x44) – key block protected using the AES key derivation binding method (see subclause 7.2.1) ‘E’ (0x45) – key block protected using the AES key derivation binding method for CTR mode. Note that numeric key block version IDs are reserved for proprietary key block definitions. Also note that multiple key block versions may be in use at any time. The key block version ID should not be used as a proxy for determining which optional block elements are supported. Applications should predetermine which optional block elements are supported. Future versions of this document might define other values. The values ‘A’, ‘B’ and ‘C’ are not used by this document.	1AN	No

Table 1 (continued)

Byte #	Field name	Description	Encoding	Encrypted
1-4	Key block length	ASCII numeric digits providing key block length after encoding. Length includes the entire block (header + encrypted confidential data + MAC) in decimal; for example, a 112-byte key block would contain '0' (0x30) in byte #0, '1' (0x31) in byte #1, '1' (0x31) in byte #2 and '2' (0x32) in byte #3	4N	No
5-6	Key usage	Provides information about the intended function of the protected key or sensitive data. Common functions include encrypting data, encrypting PINs and calculating a MAC. See Table 2 for defined values.	2AN	No
7	Algorithm	The approved algorithm for which the protected key may be used. See Table 3 for defined values.	1AN	No
8	Mode of use	Defines the operation the protected key can perform. For key blocks being transmitted, the mode of use is in the context of the recipient SCD. For key blocks being stored, the mode of use is in the context of the SCD using the key. For example, an authentication key might be limited to verify-only. See Table 4 for defined values.	1AN	No
9-10	Key version number	Two-digit ASCII character version number, optionally used to indicate that the content of the key block is a component, or to prevent re-injection of old keys. See Table 5 for defined values.	2AN	No
11	Exportability	Defines whether the protected key may be transferred outside the cryptographic domain in which the key is found. See Table 6 for defined values.	1AN	No
12-13	Number of optional blocks	Defines the number of optional blocks included in the key block. The minimum value is zero and the maximum is 99, but it is also limited by the maximum number of bytes in the entire key block. For example, 5 optional blocks would be represented by '0' (0x30) in byte 12 and '5' (0x35) in byte 13.	2N	No
14	Key context – storage or exchange	This field indicates whether the key block is in a key exchange context (wrapped by a transport key) or in a storage context (e.g. wrapped by the MFK). The key context value does not require a certain exportability setting. The values are: '0' (0x30) – Either storage or key exchange context. This allows interoperability with legacy key blocks. Note that this does not imply that the wrapping key for a key block can be used for both storage and key exchange, merely that the storage or exchange of this key block is determined by the properties of the wrapping key. '1' (0x31) – Storage context only. The key block is not wrapped by a transport key for exchange between a communicating pair. '2' (0x32) – Key exchange context only. The key block is wrapped by a transport key for exchange between a communicating pair.	1N	No
15	Reserved for future use	This field is reserved for future use and is filled with ASCII zero (0x30) characters.	1N	No
Fields after byte 15 are only present if bytes 12-13 contain a value other than '00' (0x3030)				
16-17	First optional block ID	ID field for the first optional block, if one is present as indicated by a value other than '00' in bytes 12 and 13. The possible ID values are defined in subclause 6.3.6 .	2AN	No

Table 1 (continued)

Byte #	Field name	Description	Encoding	Encrypted
18-19	optional block 1 length	If the first optional block is present, indicated by a value other than '00' in bytes 12 and 13, then this field contains the length of that optional block in bytes, including the field's ID, length and data. Byte 18 contains the high-order byte of the length and byte 19 contains the low-order byte, both in hex-ASCII form. For example, for a 24-byte optional block (0x18 in hexadecimal notation), byte 18 would contain a '1' (0x31) and byte 19 would contain an '8' (0x38). optional block data might be omitted, in which case the optional block length will be '04'. If the length of the optional block exceeds 255, this field is set to '00' and the actual length is encoded in the next two fields. The actual length is limited by the maximum number of bytes in the key block length.	2H	No
20-n	optional block 1 length of length	This field contains the length of the field named "Extended optional block 1 length". Valid values are '01' – 'FF'. For this document this value is always '04'. This field is only present if the optional block 1 length is '00'.	2H	No
	Extended optional block 1 length	The actual length of the block, expressed as a hexadecimal number. For example, if the optional block length is 400 ₁₀ bytes (190 ₁₆), this field is '0190'. This field is only present if the optional block 1 length is '00'.	4H	No
n-m	optional block 1 data	If the first optional block is present, indicated by a value other than '00' in bytes 12-13, then this field contains the data for that optional block. It will contain only printable ASCII characters.	PA	No
n-m	Additional optional blocks, if present	A variable number of optional blocks can follow the first optional block, as indicated by the count in bytes 12-13. Each optional block contains a 2-byte ID, 2-byte length and variable-length data field following the same format described for optional block 1.		No

Before a key in the key block format is used in a secure cryptographic device (SCD), the contents of the header block shall be validated to ensure the correct usage is enforced. First, the SCD makes sure that the length of the key block matches the contents of bytes 1 to 4. If the lengths do not match, the SCD returns an error and stops processing the block. The "key usage" byte is typically checked next followed by the "algorithm" byte. The processing of the other header bytes depends on the key usage and the algorithm used.

From 0-99 optional blocks can be included in the key block, with the limitation that the total length of the key block shall not exceed the maximum length that can be represented by the key block length field in bytes 1-4. The total number of optional blocks is indicated by a count in bytes 12-13. Each optional block consists of three fields: a 2-byte identifier that indicates the content of the optional block, a 2-byte length that indicates the total number of bytes in all of the optional block's fields and the variable-length data carried in the optional block. See [Table 7](#) for a list of valid identifiers.

As an example, consider an optional block that contains a key set identifier. A sample block could contain the following data.

Identifier: 'KS'

Length: '18'

Data: '00604B120F9292800000'

NOTE The length value equals 2+2+20 (24). The length value is encoded in Hex-ASCII; 18 hexadecimal equals 24 decimal.

If more than one optional block is present, each one shall immediately follow the previous one in the key block structure.

The total length of all optional blocks in the key block shall be a multiple of the encryption block size. This might require padding, and if padding is needed it shall be included in a special final optional block that is filled with an appropriate number of padding characters. For example, if the total length of all optional blocks before padding is 90, six additional bytes are needed in order to bring the total to the next multiple of the encryption block size, 96. A padding block would be added as the last optional block, containing the 2-byte identifier 'PB', a 2-byte length field '06' and two characters of padding.

6.2.2 Changing Key Headers

A single key under two different headers with different attributes creates the potential for attacks. Therefore, the SCD shall not generate a key with multiple headers or change the value of a key header except as described in the list after this paragraph. This includes changing the header when changing the wrapping key. Similarly, there shall be a policy that users shall not manually load the same key bound to different attributes except under the same rules. Checking manual loading rules is generally not enforceable within the HSM since these are distributed systems. SCDs that translate a key to a non- ANSI X9.143/TR-31 format should maintain equivalent usage separation to the ANSI X9.143/TR-31 headers.

Legitimate changes to the headers are as follows:

- a) A header may be changed to become more restrictive. The reverse operation of any of these is not allowed (e.g. an encrypt-only key cannot become both encrypt and decrypt).
 - 1) An exportable key may become non-exportable (i.e. exportability "E" may become "N").
 - 2) A key that supports both encrypt and decrypt can become either encrypt-only or decrypt-only (i.e. mode of use "B" can become mode of use "E" or "D").
 - 3) A key that supports both generate and verify can become generate-only or verify-only (i.e. mode of use "C" can become mode of use "G" or "V").
 - 4) A key that supports both sign and decrypt can become sign-only or decrypt-only (i.e. mode of use "T" can become mode of use "S" or "D").
- b) SCDs may generate a key marked as non-exportable and export that key in a single key export command or operation. Effectively, this is shorthand for internally generating an exportable version of that key, exporting the key as non-exportable key and then changing the local copy of the key to non-exportable. Since the original exportable version of the key is never exposed outside of the SCD, this creates a "one-time exportable" key. The SCD shall not export keys marked as non-exportable except as part of the key generation function.
- c) Uni-directional keys need to occur in matching pairs. For example, an encrypt-only key at the sender needs a decrypt-only (or both direction) key at the receiver. This means that reversing the mode of use upon export can be necessary. However, this creates a potential weakness in that an SCD's commands could be used with a bi-directional KBPK to change the usage of the key within the SCD.
 - 1) Keys that have strong single-direction usage requirements shall be marked non-exportable.
 - 2) Single-direction keys that are exportable or one-time exportable should only be exported under single-direction KBPKs.
- d) optional blocks may be added, removed or changed, except for the HMAC optional block, which cannot be removed or changed when included.
- e) Keys with usage "K0" may be converted to be keys with usage "K1".

6.3 Defined values for key block headers

6.3.1 Key Usage

Key usage defines the type of the key, whether it is used, for example, to encrypt data or calculate a MAC. The key usage is identified by bytes 5 and 6. [Table 2](#) shows the currently defined key usage values. There is additional information for each defined value after [Table 2](#).

Table 2 — Defined key usage values

Value	Hex	Definition	Mode(s) of Use	Allowed key block header version ID values
'B0'	0x4230	Base derivation key (BDK)	'X'	'D','E'
'B1'	0x4231	Initial DUKPT key	'X'	'D','E'
'B2'	0x4232	Base key variant key (deprecated)	'Y'	'D','E'
'B3'	0x4233	Key derivation key (non-ANSI X9.24 / ISO 11568)	'X'	'D','E'
'B4'	0x4234	Key diversification key Type A (key diversification scheme, see Annex A)	'X'	'D','E'
'B5'	0x4235	Key diversification key Type B (key diversification scheme, see Annex A)	'X'	'D','E'
'C0'	0x4330	Card verification key (CVK)	'C', 'G', 'V'	'D','E'
'D0'	0x4430	Symmetric key for data encryption	'B', 'D', 'E'	'D','E'
'D1'	0x4431	Asymmetric key for data encryption	'B', 'D', 'E'	'D','E'
'D2'	0x4432	Data encryption key for decimalization table	'B', 'D', 'E'	'D','E'
'D3'	0x4433	Data encryption key for sensitive data	'B', 'D', 'E'	'D','E'
'E0'	0x4530	EMV (Europay MasterCard and Visa ICC specification)/chip issuer master key: application cryptograms	'X'	'D','E'
'E1'	0x4531	EMV/chip issuer master key: secure messaging for Confidentiality	'X'	'D','E'
'E2'	0x4532	EMV/chip issuer master key: secure messaging for Integrity	'X'	'D','E'
'E3'	0x4533	EMV/chip issuer master key: data authentication Code	'X'	'D','E'
'E4'	0x4534	EMV/chip issuer master key: dynamic numbers	'X'	'D','E'
'E5'	0x4535	EMV/chip issuer master key: card personalization	'X'	'D','E'
'E6'	0x4536	EMV/chip issuer master key: other	'X'	'D','E'
'E7'	0x4537	EMV/Chip Asymmetric key pair for EMV/smart-card-based PIN/PIN block encryption	'B', 'D', 'E'	'D','E'
'F0'	0x4630	EMV/chip card key: application cryptograms	'X'	'D','E'
'F1'	0x4631	EMV/chip card key: secure messaging for Confidentiality	'X'	'D','E'
'F2'	0x4632	EMV/chip card key: secure messaging for integrity	'X'	'D','E'
'F3'	0x4633	EMV/chip card key: data authentication code	'X'	'D','E'
'F4'	0x4634	EMV/chip card key: dynamic numbers	'X'	'D','E'
'F5'	0x4635	EMV/chip card key: card personalization	'X'	'D','E'
'F6'	0x4636	EMV/chip card key: other	'X'	'D','E'
'I0'	0x4930	Initialization vector (IV)	'N'	'D','E'
'K0'	0x4B30	Key encryption key or key wrapping key	'B', 'D', 'E'	'D','E'
'K1'	0x4B31	KBPK, ANSI X9.143/TR-31	'B', 'D', 'E'	'D','E'

Table 2 (continued)

Value	Hex	Definition	Mode(s) of Use	Allowed key block header version ID values
'K2'	0x4B32	Asymmetric key pair (KRD), ANSI X9.139/TR-34	'B', 'D', 'E'	'D', 'E'
'K3'	0x4B33	Asymmetric key pair for key wrapping/key Agreement	'B', 'D', 'E', 'X'	'D', 'E'
'K4'	0x4B34	KBPK, ISO 20038	'B', 'D', 'E',	'D', 'E'
'M0'	0x4D30	MAC key, ISO 16609 MAC algorithm 1 (using TDEA)	'C', 'G', 'V'	'D', 'E'
'M1'	0x4D31	MAC key, ISO/IEC 9797-1 MAC algorithm 1	'C', 'G', 'V'	'D', 'E'
'M2'	0x4D32	MAC key, ISO/IEC 9797-1 MAC algorithm 2	'C', 'G', 'V'	'D', 'E'
'M3'	0x4D33	MAC key, ISO/IEC 9797-1 MAC algorithm 3	'C', 'G', 'V'	'D', 'E'
'M4'	0x4D34	MAC key, ISO/IEC 9797-1 MAC algorithm 4	'C', 'G', 'V'	'D', 'E'
'M5'	0x4D35	MAC key, ISO/IEC 9797-1:1999 MAC algorithm 5	'C', 'G', 'V'	'D', 'E'
'M6'	0x4D36	MAC key, ISO/IEC 9797-1:2011 MAC algorithm 5/CMAC	'C', 'G', 'V'	'D', 'E'
'M7'	0x4D37	HMAC key	'C', 'G', 'V'	'D', 'E'
'M8'	0x4D38	MAC key, ISO/IEC 9797-1:2011 MAC algorithm 6	'C', 'G', 'V'	'D', 'E'
'P0'	0x5030	PIN encryption key	'B', 'D', 'E'	'D', 'E'
'P1'	0x5031	PIN generation key (reserved for ANSI X9.132-202x)	'G'	'D', 'E'
'P2'	0x5032	PIN generation key, other algorithm	'G'	'D', 'E'
'S0'	0x5330	Asymmetric key pair for digital signature	'S', 'V'	'D', 'E'
'S1'	0x5331	Asymmetric key pair, CA key	'S', 'V'	'D', 'E'
'S2'	0x5332	Asymmetric key pair, non-ANSI X9.24/ ISO 11568 key	'S', 'V', 'T', 'B', 'D', 'E'	'D', 'E'
'V0'	0x5630	PIN verification key, PVK, other algorithm	'C', 'G', 'V'	'D', 'E'
'V1'	0x5631	PIN verification key, IBM 3624	'C', 'G', 'V'	'D', 'E'
'V2'	0x5632	PIN verification key, VISA PVV	'C', 'G', 'V'	'D', 'E'
'V3'	0x5633	PIN verification key, ANSI X9.132 algorithm 1	'C', 'G', 'V'	'D', 'E'
'V4'	0x5634	PIN verification key, ANSI X9.132 algorithm 2	'C', 'G', 'V'	'D', 'E'
'V5'	0x5635	PIN verification key, ANSI X9.132 algorithm 3	'C', 'G', 'V'	'D', 'E'
All numeric values		Reserved for proprietary use (acceptable modes of use vary depending on rules of use for each such key)		

Note that some of these usages can work for both symmetric and asymmetric keys. Usage 'K0' is appropriate for a TDEA KEK and an RSA key exchange key. The key usage 'K0' should not be used for asymmetric keys in new implementations; key usage 'K3' should be used.

For new implementations, mode of use values for asymmetric private keys should specify appropriate usages (e.g. the 'E' and 'B' mode of use values are not appropriate for a private key).

This list of usages is not intended to be exhaustive. Future editions could include other key usage types.

Precise usage restrictions for each of these usage values are described in the next paragraphs. The SCD shall ensure the key usages are enforced according to the restrictions identified.

BDK base derivation key (B0) – This key shall only be used to derive the initial DUKPT key ('B1') in the DUKPT process defined in ANSI X9.24-3/ ISO 11568. This key Type shall only be used in conjunction with the Derivation key usage, 'X'.

Initial DUKPT key (B1) – This key shall only be used as the initial DUKPT key defined for the DUKPT key management scheme in ANSI X9.24-3/ ISO 11568. This key type shall only be used in conjunction with the Derivation key usage, 'X'.

Base key variant key (B2) – This key is used to create key variants from the base key variant key. It shall not be used in any other key variant process. The variant keys shall only be created and used inside the SCD that holds the base key variant key. This key type shall only be used in conjunction with the Derivation key usage, 'Y'. Note that the (B2) usage is deprecated.

Key derivation key (B3) – This key is used as an input to an irreversible key derivation function to derive other keys. It shall not be used in any key derivation process defined in ANSI X9.24/ ISO 11568. For example, this key might be used as input to a key derivation function defined in NIST SP800-108. This key type shall always be used in conjunction with the Derivation key usage, 'X'. This key type should be accompanied by a proprietary algorithm ('PA') optional block that specifies the algorithm.

Key diversification key type A (B4) This key shall only be used as a type A input to the irreversible key diversification scheme, see [Annex A](#). This key type shall always be used in conjunction with the Derivation key usage, 'X'. This key usage shall be used with the derivation allowed ('DA') optional block version '02'. This key type should be accompanied by a proprietary algorithm ('PA') optional block that specifies the derivation algorithm when the m value in the DA optional block is 1 or greater..

Key diversification key type B (B5) This key shall only be used as a type B input to the irreversible key diversification scheme, see [Annex A](#). This key type shall always be used in conjunction with the Derivation key usage, 'X'. This key usage shall be used with the derivation allowed ('DA') optional block version '02'. This key type should be accompanied by a proprietary algorithm ('PA') optional block that specifies the derivation algorithm when the m value in the DA optional block is 1 or greater..

CVK card verification key (C0) – This key is used to compute either a static or dynamic card verification code such as those using the CVV, CVC, CSC, CVC2 and CVV2 algorithms. It can also be used to verify those codes, which consists of computing the code and comparing the result to a supplied value.

Symmetric key for data encryption (D0) – This key is used to encipher or decipher data.

Asymmetric key for data encryption (D1) – The public and private key pair used to encipher or decipher data (e.g. encrypting a PAN using RSA or Elliptic Curve ECIES). If ECIES is used, the derived encryption and message authentication keys are ephemeral and shall not be disclosed outside the SCD. Note that the D1 key usage value should not be used when wrapping a symmetric key in a digital envelope, even if the symmetric key is not stored or returned. Asymmetric keys use for key protection should be tagged as 'K2' or 'K3'.

Data encryption key for Decimalization Table (D2) – This key is used to encipher or decipher a decimalization table.

Data encryption key for sensitive data (D3) – This key is used to protect data (e.g. PANs or passwords). D3 keys shall not be used in data decryption commands that return the decrypted data in the clear. They shall only translate the data to another D3 key, decrypt the data as part of a verification process or encrypt the data.

EMV/chip issuer master key: application cryptograms (E0) – Keys with this type are used for deriving the EMV chip card key used to produce application cryptograms (TC, ARQC or AAC). Typically used by an issuer to generate the corresponding F0 key.

EMV/chip issuer master key: secure messaging for confidentiality (E1) – Keys with this type are used for deriving the EMV chip card master key used to ensure messaging confidentiality. Typically used by an issuer to generate the corresponding F1 key.

EMV/chip issuer master key: secure messaging for integrity (E2) – Keys with this type are used for deriving the EMV chip card master key used to ensure messaging integrity. Typically used by an issuer to generate the corresponding F2 key.

EMV/chip issuer master key: data authentication code (E3) – Keys with this type are used for deriving the EMV chip card master key used to produce the DAC. Typically used by an issuer to generate the corresponding F3 key.

EMV/chip issuer master key: dynamic numbers (E4) – Keys with this type are used for deriving the EMV chip card master key used for producing Dynamic numbers. Typically used by an issuer to generate the corresponding F4 key.

EMV/chip issuer master key: card personalization (E5) – Keys with this type are used for deriving the EMV chip card master key used to protect card data during personalization. Typically used by an issuer to generate the corresponding F5 key.

EMV/chip issuer master key: other (E6) – Keys with this type are used for deriving the EMV chip card master key used for any other purposes. Typically used by an issuer to generate the corresponding F6 key.

EMV/chip asymmetric key pair for EMV/smart card based PIN/PIN block encryption (E7) – Keys with this type are used for loading PINs on to EMV chip cards.

Note that requirements for the use, generation and management of EMV keys are defined by the specifications written by EMVCo and by the card associations that make use of EMV cards. These requirements are not defined in standards developed by ASC X9.

EMV/chip card key: application cryptograms (F0) – Keys with this type are used for the EMV chip card key used to produce application cryptograms (TC, ARQC or AAC).

EMV/chip card key: secure messaging for confidentiality (F1) – Keys with this type are used for the EMV chip card master key used to ensure messaging confidentiality.

EMV/chip card key: secure messaging for integrity (F2) – Keys with this type are used for the EMV chip card master key used to ensure messaging integrity.

EMV/chip card key: data authentication code (F3) – Keys with this type are used for the EMV chip card master key used to produce the DAC.

EMV/chip card key: dynamic numbers (F4) – Keys with this type are used for the EMV chip card master key used for producing Dynamic numbers.

EMV/chip card key: card personalization (F5) – Keys with this type are used for the EMV chip card master key used to protect card data during personalization.

EMV/chip card key: other (F6) – Keys with this type are used for the EMV chip card master key used for any other purposes.

Initialization vector (I0) – This value is used as input to any cipher mode of operation requiring an initialization vector (IV) or counter value (e.g. MAC, data encryption or decryption). It is not a key and shall not be used for any purpose other than as an initialization value. Note that key usage appropriate for the IV is 'N' since the IV is not a key. The algorithm for which the IV can be used is defined by the algorithm identifier in the header.

Key encryption key or key wrapping key (K0) – This is a key encryption key (KEK) or key wrapping key (KWK) which shall be used only to encrypt or decrypt other keys, or as a key used to derive keys that are used for that purpose. Note that this key usage should not be used for ANSI X9.143/TR-31 KBPKs. When creating new keys for a communicating pair that will be used to create ANSI X9.143/TR-31 defined key blocks, the new key shall be designated as a KBPK (K1). Legacy KBPK KEKs with a key usage of "K0" under TR-31 version "A" should be converted to key usage as "K1" (see [subclause 6.2.2](#)).

Key block protection key (K1) – The derivation key from which the KBK and the KBAK are derived; this key shall only be used for this purpose. Keys tagged as "K1" shall not be re-tagged as "K0".

Asymmetric key pair (KRD), ANSI X9.139/TR-34 (K2) – The asymmetric public and private key pair used to encipher and decipher the ephemeral encryption key used with ANSI X9.139/TR-34 for key exchange.

Asymmetric key pair for key wrapping/key agreement (K3) – The asymmetric public and private key pair used for key wrapping or key agreement; this key pair shall not be used with ANSI X9.139/TR-34 for key exchange. Keys tagged as "K3" shall not be re-tagged as "K2".

Key block protection key (K4) – Deprecated, do not use in new implementations. Previously defined as an AES KBPK to be used with this document. New implementations should use key usage “K1”.

MAC key, ISO 16609 MAC algorithm 1 (using TDEA) (M0) – This key can be used to either compute (generate) or verify a MAC using the ISO 16609 algorithm 1. This key shall not be used with any other MAC algorithms.

MAC key, ISO/IEC 9797-1 MAC algorithm 1 (M1) – This key can be used to either compute (generate) or verify a MAC using the ISO/IEC 9797-1 algorithm 1. This key shall not be used with any other MAC algorithms.

MAC key, ISO/IEC 9797-1 MAC algorithm 2 (M2) – This key can be used to either compute (generate) or verify a MAC using the ISO/IEC 9797-1 algorithm 2. This key shall not be used with any other MAC algorithms.

MAC key, ISO/IEC 9797-1 MAC algorithm 3 (M3) – This key can be used to either compute (generate) or verify a MAC using the ISO/IEC 9797-1 algorithm 3. This key shall not be used with any other MAC algorithms.

MAC key, ISO/IEC 9797-1 MAC algorithm 4 (M4) – This key can be used to either compute (generate) or verify a MAC using the ISO/IEC 9797-1 algorithm 4. This key shall not be used with any other MAC algorithms.

MAC key, ISO/IEC 9797-1:1999 MAC algorithm 5 (M5) – This key can be used to either compute (generate) or verify a MAC using the ISO/IEC 9797-1:1999 algorithm 5. This key shall not be used with any other MAC algorithms.

MAC key, ISO/IEC 9797-1:2011 MAC algorithm 5 (M6) – This key can be used to either compute (generate) or verify a MAC using the ISO/IEC 9797-1:2011 algorithm 5. This algorithm is the same as CMAC as defined in NIST SP800-38B. This key shall not be used with any other MAC algorithms.

HMAC key (M7) – This key can be used to either compute (generate) or verify an HMAC, as specified in FIPS 198-1, using a cryptographic key in conjunction with a hash function. This key shall not be used with any other MAC algorithms.

MAC key, ISO/IEC 9797-1:2011 MAC algorithm 6 (M8) – This key can be used to either compute (generate) or verify a MAC using the ISO/IEC 9797-1 algorithm 6. This key shall not be used with any other MAC algorithms.

PIN encryption key (P0) – This key is used to protect PIN blocks. It can be used with any PIN block format and with any PIN block encryption method. This key shall not be used with any type of data other than a PIN or PIN block.

PIN generation key (P1) – This key is reserved for generation of PINs according to ANSI X9.132-202x. This key shall not be used for any other purpose.

PIN generation key, other algorithm (P2) – A key of this type is used for generation of PINs according to algorithms other than ANSI X9.132-202x. This key shall not be used for generation of PINs according to ANSI X9.132-202x, for PIN verification or for any function other than PIN generation. This key type should be accompanied by a proprietary algorithm ('PA') optional block that specifies the algorithm.

Asymmetric key pair for digital signature (S0) – Asymmetric public or private keys used for computing and verifying digital signatures.

Asymmetric key pair, CA key (S1) – Asymmetric public or private keys used by a CA to compute and verify digital signatures for certificates.

Asymmetric key pair, non-ANSI X9.24 key/ ISO 11568 (S2) – Asymmetric public or private keys used for computing and verifying digital signatures and used to encrypt or decrypt other keys. S2 keys are not compliant with ANSI X9.24 part 2 / ISO 11568 and shall not manage keys based on ANSI X9.24/ ISO 11568 (e.g. TLS Server keys are used for key agreement, though some TLS Server keys are also authorized for signature. This key type supports mode of use “T”, both sign and decrypt).

PIN verification key, PVK, other algorithm (V0) – A key of this type is used to verify a PIN using an algorithm other than IBM 3624 or Visa PVV, each of which have their own specific key types. These keys shall not be used for the IBM 3624 or Visa PVV verification algorithms or with any function other than PIN

verification. This key type should be accompanied by a proprietary algorithm ('PA') optional block that specifies the algorithm.

PIN verification key, IBM 3624 (V1) – A key of this type is used in the IBM 3624 PIN offset generation process or to verify a PIN using the IBM 3624 algorithm. This key shall not be used to verify a PIN with any other algorithm or with any function other than PIN verification.

PIN verification key, VISA PVV (V2) – A key of this type is used to generate the PIN verification value (PVV) or to verify a PIN using the Visa PVV algorithm. This key shall not be used to verify a PIN with any other algorithm or with any function other than PIN verification.

PIN verification key, ANSI X9.132 algorithm 1 (V3) – A key of this type is used to verify a PIN using an AES algorithm. These keys shall not be used for the IBM 3624 or Visa PVV verification algorithms, or with any function other than PIN verification.

PIN verification key, ANSI X9.132 algorithm 2 (V4) – A key of this type is used to verify a PIN using an AES algorithm. These keys shall not be used for the IBM 3624 or Visa PVV verification algorithms, or with any function other than PIN verification.

PIN verification key, ANSI X9.132 algorithm 3 (V5) – A key of this type is used to verify a PIN using an AES algorithm. These keys shall not be used for the IBM 3624 or Visa PVV verification algorithms, or with any function other than PIN verification.

6.3.2 Algorithm

The algorithm byte defines what algorithm can be used with this key. The algorithm is described by byte 7 of the header. [Table 3](#) shows the currently defined algorithm values.

Table 3 — Defined algorithm values

Value	Hex	Definition	Allowed key block header version ID values
'A'	0x41	AES	'D','E'
'D'	0x44	DEA (note that this is included for backwards compatibility)	'D','E'
'E'	0x45	Elliptic Curve	'D','E'
'H'	0x48	HMAC, as specified in FIPS 198-1, (specify the underlying hash algorithm in optional field)	'D','E'
'M'	0x4D	SM4, as specified in ISO/IEC 18033-3:2010/Amd 1:2021	'D','E'
'R'	0x52	RSA	'D','E'
'S'	0x53	DSA (note that this is included for future reference)	'D','E'
'T'	0x54	Triple DEA (TDEA)	'D','E'
Numeric values		Reserved for proprietary use	

6.3.3 Mode of use

The mode of use byte shall define the operation the key can perform on the system where the key is used. The mode is identified in byte 8 of the KBH. For asymmetric keys this mode shall be applied in the context of the private key. [Table 4](#) shows the currently defined mode of use values.

Table 4 — Defined mode of use values

Value	Hex	Definition	Allowed key block header version ID values
'B'	0x42	Both encrypt / wrap and decrypt / unwrap	'D','E'
'C'	0x43	Both generate and verify	'D','E'
'D'	0x44	Decrypt/unwrap only	'D','E'
'E'	0x45	Encrypt/wrap only	'D','E'
'G'	0x47	Generate only	'D','E'
'N'	0x4E	No special restrictions (other than restrictions implied by the key usage)	'D','E'
'S'	0x53	Signature only	'D','E'
'T'	0x55	Both sign and decrypt	'D','E'
'V'	0x56	Verify only	'D','E'
'X'	0x58	Key used to derive other key(s)	'D','E'
'Y'	0x59	Key used to create key variant(s)	'D','E'
Numeric values		Reserved for proprietary use	

Precise usage restrictions for each of these mode values are described as follows. The SCD shall ensure the mode of use is enforced according to the restrictions identified in the next paragraphs.

Both encrypt/wrap and decrypt/unwrap (B) – The key can be used in encryption and decryption, wrapping and unwrapping. This includes a variety of key usages and not only data encryption and decryption.

Both generate and verify (C) – The key can be used to generate and verify a MAC. This mode is compatible with keys having authentication key usage attributes, as well as other attributes in which the key is used to compute and verify a MAC

Decrypt/unwrap only (D) – The key can be used in a decryption process. This includes a variety of key usages and not only data decryption. This mode indicates that the key can be used for unwrapping but cannot be used for wrapping. If the key usage is 'P0', PIN encryption key, a mode of use value of "decrypt/unwrap only" means that this key is only used to decrypt PINs as the first half of a translate or in preparation for a verify operation.

Encrypt/wrap only (E) – The key can be used in an encryption process. This includes a variety of key usages and not only data encryption. This mode indicates that the key can be used for wrapping but cannot be used for unwrapping. If the key type is 'P0', "PIN encryption key", a mode of use value of "Encrypt/Wrap only" may be used to encrypt a PIN in a PED or as the second half of a PIN translate operation. This usage only applies to symmetric keys. For asymmetric keys this mode is applied in the context of the private key.

Generate only (G) – The key can be used to generate a verification value (e.g. MAC or PIN Offset), but it cannot be used to verify the value.

No special restrictions (N) – This mode is used when no other modes are applicable. This value should only be used with IVs and for proprietary key usages.

Signature only (S) – The key can only be used in digital signature operations. This means that the key shall be a public key (signature verify) or a private key (signature generate) from a public-key algorithm such as RSA or elliptic curve digital signature (e.g., ECDSA).

Both sign and decrypt (T) – The key can be used to sign and decrypt. Not to be used for keys in scope of ISO 11568.

Verify only (V) – The key can be used to verify a value (e.g. MAC or PIN Offset), but it cannot be used to generate the value.

Key used to derive other key(s) (X) – The key is used only in a key derivation process that produces one or more derived keys.

Key used to create key variant(s) (Y) – The key is used only in a process that produces one or more variants of a key [see definition ‘base key variant key (B2)’ in [subclause 6.3.1](#)].

6.3.4 Key Version Number

The key version number is a 2-byte field that can be used either to store key version number or to indicate the value is a component of a key. The key version number shall have the values defined in [Table 5](#).

Table 5 — Key version number definition

First character	Second character	Key value field meaning
‘0’ (0x30)	‘0’ (0x30)	Key versioning is not used for this key.
‘c’ (0x63)	Any character	The value carried in this key block is a component of a key. An organization's policy will dictate the proper use of a component. Typically, a key with a specific header will be derived from two or more components with the same header (with the obvious exception of the key version bytes).
Any other combination of characters		The key version field indicates the version of the key carried in the key block. The value may optionally be used to prevent re-injection of old keys.

This field should not be confused with byte 0, the key block version ID. Byte 0 indicates which version of the key block description is being used. In contrast, the key version number is a tool for enforcement of local key change rules.

6.3.5 Exportability

“Exportability” refers to transfer outside the cryptographic domain in which the key is found. Secure backup of a key may be allowed, depending on the organization's policy. The value of the exportability byte does not preclude the ability for the SCD to translate the key block under a storage key. However, translation of a key block stored under a storage key of a cryptographic domain to a storage key from another cryptographic domain shall be considered an export of the key block and shall be governed by the exportability setting of the key block.

Flags in this field indicate special types of keys that require unusual handling. Any key that does not follow normal security assumptions and does not have an E, N or S shall have a notation in the reserved field.

Exportability is byte 11 of the header. [Table 6](#) shows the currently defined values for the exportability byte. The SCD shall ensure the key exportability is enforced according to the restrictions identified in [Table 6](#). See [Annex A](#) for operational challenges and examples of use for the exportability value.

Table 6 — Defined values for exportability Byte

Value	Hex	Definition
‘E’	0x45	Exportable under a KEK in a form meeting the requirements of ISO 11568.
‘N’	0x4E	Non-exportable by the receiver of the key block or from storage. Does not preclude exporting keys derived from a non-exportable key.
‘S’	0x53	Sensitive, exportable under a KEK in a form not necessarily meeting the requirements of ISO 11568. Examples include: export of a key under a KEK that is weaker than itself, or export of a key using a mechanism that does not retain the key attributes specified in the key block. Examples such as these are non-compliant with ISO 11568, but are allowed if a key block uses the ‘S’ exportability setting.
Numeric values		Reserved for proprietary use

NOTE 1 Exportability has nothing to do with government regulations.

6.3.6 Optional block ID

6.3.6.1 General

Each optional block shall begin with a 2-byte identifier that indicates the type of optional data. [Table 7](#) shows the currently defined values for the optional block ID. If an SCD encounters an optional block ID value that it does not understand, it should reject the key block containing that optional block.

An optional block ID value shall not be repeated within an ANSI X9.143 key block. Other than optional blocks for HMACs, information in optional fields is not necessarily SCD enforced.

[Figure 3](#) shows simplified examples of the three possible configurations of the key block header, in terms of optional blocks.

- The left part of the diagram shows the case when there are no optional blocks. In this case, the “number of optional blocks” in the header is set to zero by making its value ASCII ‘00’ (0x3030). No optional block information at all is appended to the base key block header structure.
- The centre of the diagram shows the case where optional blocks are present, and the total length of all optional blocks combined happens to be a multiple of the encryption block length. In this specific example, two optional blocks are shown. In this case, the presence of two optional blocks is shown by the value 2 (ASCII ‘02’ 0x3032) in the “number of optional blocks”, and the information for the two optional blocks immediately follows the base key block header structure. Note that the order of the optional blocks is unconstrained in this case; they can appear in any order.
- The right part of the diagram shows the case where the creator of the key block needs to include optional blocks, but the total length of those optional blocks combined is not a multiple of the encryption block length. In the example, the creator needs to include two optional blocks, shown as “first optional block” and “Second optional block”. Padding is required because the length of the key block header and optional blocks is not a multiple of the encryption block length. The padding is included by appending a third optional block called the padding block to the two optional blocks that were required for functional purposes. The padding block is formed so that its total length, when added to the combined total lengths of all other optional blocks in the header, is a multiple of the encryption block length. If used, the padding block will always be the last optional block in the header. There is no constraint on the order of the optional blocks that may precede it in the header; they can be in any order.

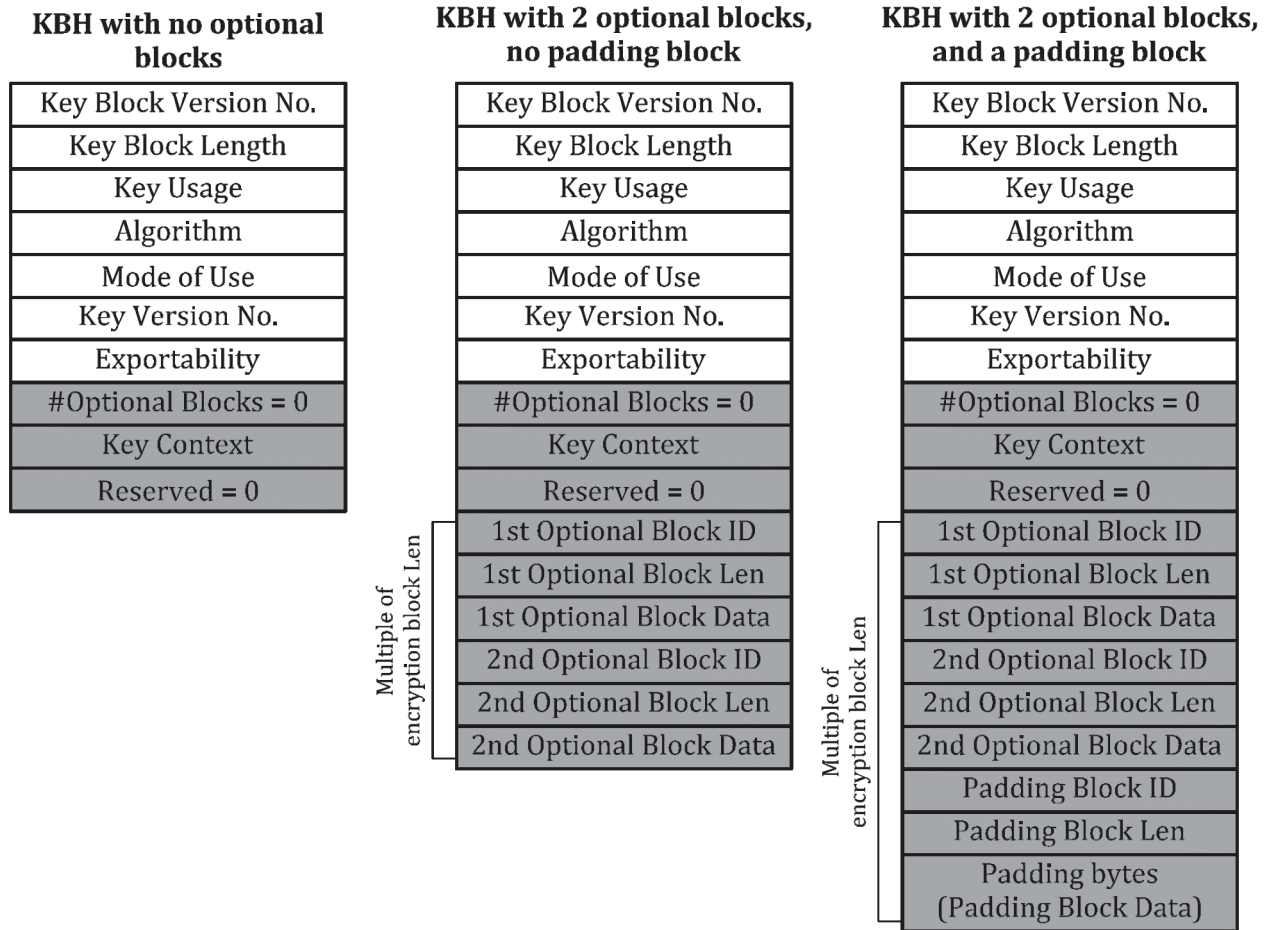


Figure 3 — Examples of KBH and optional blocks

[Table 7](#) shows the currently defined values for the optional block ID.

Table 7 — Defined values for optional block ID

Value	Hex	Definition
'AL'	0x414C	Asymmetric key life attribute.
'BI'	0x4249	Base derivation key identifier for DUKPT
'CT'	0x4354	Asymmetric public key certificate or chain of certificates.
'DA'	0x4441	Derivation(s) allowed for derivation keys
'HM'	0x484D	Hash algorithm for HMAC
'IK'	0x494B	Initial key: identifier for the Initial AES DUKPT key. This identifier shall not be used for the TDEA DUKPT initial key ID - see the 'KS' optional block ID.
'KC'	0x4B43	Key check: key check value of the wrapped key; computed according to ANSI X9.24-1-2017 Annex A/ ISO 11568; the KCV is used for key validation not as an integrity mechanism (see subclause 6.3.6.8).
'KP'	0x4B50	Key protection: key check value of the KBPK; computed according to ANSI X9.24-1-2017 Annex A/ ISO 11568; the KCV is used for key validation not as an integrity mechanism (see subclause 6.3.6.8).
'KS'	0x4B53	Key serial: initial key serial number as used in TDEA DUKPT. This identifier shall not be used for the AES DUKPT initial key ID - see the 'IK' optional block ID.