



BSI Standards Publication

Financial services — Key management (retail)

National foreword

This British Standard is the UK implementation of [ISO 11568:2023](#). It supersedes [BS ISO 11568-4:2007](#), [BS ISO 11568-2:2012](#), and [BS ISO 11568-1:2005](#), which are withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/12, Financial services.

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 2023
Published by BSI Standards Limited 2023

ISBN 978 0 580 98947 6

ICS 35.240.40

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 March 2023.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

INTERNATIONAL STANDARD

ISO
11568

First edition
2023-02-17

Financial services — Key management (retail)

Services financiers — Gestion de clés (services aux particuliers)



Reference number
ISO 11568:2023(E)
Documents
Document Sharing Hub
© ISO 2023



COPYRIGHT PROTECTED DOCUMENT

© ISO 2023, Published in Switzerland

All rights reserved. Unless otherwise specified, 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
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword	v
Introduction	vi
1 Scope	1
1.1 General	1
1.2 Scope exclusions	1
2 Normative references	1
3 Terms and definitions	2
4 Key management requirements	12
4.1 General	12
4.1.1 Key management strategy	12
4.1.2 Dual control and split knowledge of secret or private keys	12
4.1.3 Permissible key forms	12
4.1.4 Logging	14
4.1.5 Cryptographic strength	14
4.1.6 Key locations	15
4.1.7 Single-purpose key usage	15
4.2 Secure cryptographic device	17
4.2.1 General requirements	17
4.2.2 Additional SCD requirements for devices used in SKDAT	18
4.3 Additional CA requirements	19
4.4 Additional RA requirements	19
4.5 Key blocks	20
4.5.1 Overview of key blocks	20
4.5.2 Key attributes	21
4.5.3 Integrity of the key block	21
4.5.4 Key and sensitive attributes field	21
4.6 Key creation	22
4.6.1 Symmetric key creation	22
4.6.2 Asymmetric key creation	23
4.7 Key component and key share creation	23
4.8 Check values	24
4.8.1 Introduction	24
4.8.2 Symmetric key check value calculation	24
4.8.3 Asymmetric key check value calculation	25
4.9 Key distribution	25
4.9.1 Symmetric key distribution	25
4.9.2 SKDAT asymmetric key distribution	28
4.10 Key loading	30
4.10.1 General	30
4.10.2 Loading key components or shares	31
4.11 Key utilization	31
4.11.1 General key utilization requirements	31
4.11.2 Additional key utilization requirements for SKDAT	32
4.12 Key storage	33
4.12.1 Cleartext key component and share storage	33
4.12.2 Public key storage	33
4.13 Key replacement	33
4.14 Key destruction	34
4.14.1 General	34
4.14.2 Key destruction from an SCD	35
4.14.3 Destruction of a key in cryptogram form	35
4.14.4 Component and share destruction	35
4.15 Key backup	35

4.16	Key archiving.....	36
4.17	Key compromise.....	36
5	Transaction key management techniques.....	38
5.1	General.....	38
5.2	Method: master keys or transaction keys.....	38
5.3	Derived unique key per transaction.....	38
5.3.1	General.....	38
5.3.2	DUKPT key management.....	39
5.3.3	Unique initial keys.....	41
5.3.4	AES DUKPT.....	42
5.3.5	KSN compatibility mode.....	45
5.3.6	Derived key OIDs.....	46
5.3.7	Keys and key sizes.....	46
5.3.8	Helper functions and definitions.....	47
5.3.9	Key derivation function algorithm.....	48
5.3.10	Derivation data.....	49
5.3.11	"Create Derivation Data" (local subroutine).....	50
5.3.12	Security considerations.....	51
5.3.13	Host security module algorithm.....	52
5.3.14	General.....	53
5.3.15	"Derive Initial Key".....	53
5.3.16	"Host Derive Working Key".....	54
5.3.17	Intermediate derivation key derivation data examples.....	54
5.3.18	Working key derivation data examples.....	55
5.3.19	Transaction-originating device algorithm.....	56
5.4	Host-to-host UKPT.....	61
	Annex A (informative) Key and component check values.....	63
	Annex B (normative) Split knowledge during transport.....	67
	Annex C (informative) Trust models and key establishment.....	69
	Annex D (informative) Symmetric key life cycle.....	77
	Annex E (informative) Asymmetric key life cycle phases.....	79
	Annex F (normative) Approved algorithms.....	82
	Annex G (informative) AES DUKPT pseudocode notation.....	83
	Annex H (informative) AES DUKPT test vectors.....	86
	Annex I (informative) TDEA-derived unique key per transaction.....	87
	Annex J (informative) Roles in payment environment.....	108
	Annex K (informative) Roles in symmetric key distribution using asymmetric techniques.....	111
	Bibliography.....	114

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 documents 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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO 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).

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 document cancels and replaces the former [ISO 11568](#) series, which has been technically revised.

The main changes are as follows:

- all parts of the series combined into a single document;
- fixed key no longer included in the permissible methods of transaction key management;
- required key replacement policy (see [4.13](#)) added;
- cleartext key injection removed;
- AES DUKPT introduced as a key management method.

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

Retail financial transactions are often transmitted over potentially non-secure channels, which, if exploited, can result in fraud. The vast range in value and volume of such transactions exposes participants to severe risks, which can be uninsurable. To protect against these risks, many institutions are employing encryption. The encryption algorithms used are in the public domain. The security and reliability of any process based on these algorithms is directly dependent on the protection afforded to secrets called cryptographic keys.

This document describes requirements and provides guidance for the secure management of cryptographic keys used to protect sensitive information in a retail financial services environment, for example in messages between a card acceptor and an Acquirer. Typical services in the retail financial services domain include point-of-sale (POS) debit and credit authorizations and automated teller machine (ATM) transactions. While it is designed with these environments in mind, it may also be used in unrelated applications. For example, such keys could be used for:

- encrypting Personal Identification Numbers (PIN) (see [ISO 9564-1](#));
- authenticating messages;
- encrypting other data;
- encrypting or deriving cryptographic keys;
- automated symmetric key distribution using asymmetric techniques.

Financial services — Key management (retail)

1 Scope

1.1 General

This document describes the management of symmetric and asymmetric cryptographic keys that can be used to protect sensitive information in financial services related to retail payments. The document covers all aspects of retail financial services, including connections between a card-accepting device and an Acquirer, between an Acquirer and a card Issuer, and between an ICC and a card-accepting device. It covers all phases of the key life cycle, including the generation, distribution, utilization, archiving, replacement and destruction of the keying material. This document covers manual and automated management of keying material, and any combination thereof, used for retail financial services. It includes guidance and requirements related to key separation, substitution prevention, identification, synchronization, integrity, confidentiality and compromise, as well as logging and auditing of key management events.

Requirements associated with hardware used to manage keys have also been included in this document.

1.2 Scope exclusions

This document does not specifically address internet banking services offered by an Issuer to their own customers through that financial institution's website or applications.

This document does not address using asymmetric keys to encrypt the Personal Identification Number (PIN) or any other data and does not address asymmetric keys managed with asymmetric keys.

This document is not intended to apply to the management of the keys installed in an ICC during manufacturing or the initial key established in an ICC during card personalization.

This document is not intended to address post-quantum encryption considerations. Key management using quantum technologies is out of scope of this document.

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.

ISO/IEC 9797 (all parts), *Information technology — Security techniques — Message Authentication Codes (MACs)*

ISO/IEC 11770 (all parts), *Information security — Key management*

ISO 13491 (all parts), *Financial services — Secure cryptographic devices (retail)*

[ISO 16609](#), *Financial services — Requirements for message authentication using symmetric techniques*

[ISO/IEC 18031](#), *Information technology — Security techniques — Random bit generation*

[ISO/IEC 18032](#), *Information security — Prime number generation*

ISO/IEC 18033 (all parts), *Information security — Encryption algorithms*

[ISO/IEC 19592-2](#), *Information technology — Security techniques — Secret sharing — Part 2: Fundamental mechanisms*

[ISO/IEC 19772](#), *Information security — Authenticated encryption*

[ISO 20038](#), *Banking and related financial services — Key wrap using AES*

ISO 21188:2018, *Public key infrastructure for financial services — Practices and policy framework*

ANSI X9.63, *Public Key Cryptography for the Financial Services Industry — Key Agreement and Key Management Using Elliptic Curve-Based Cryptography*

ANSI X9.143, *Retail Financial Services — Interoperable Secure Key Block Specification*

RFC 3647, *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, *Internet Request for Comments 3647*, S. Chokhani, W. Ford, R. Sabett, C. Merrill, S. Wu, November 2003

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

Acquirer

institution (or its agent) which acquires from the card acceptor the data relating to the *transaction* (3.84) and initiates the data into an *Interchange* (3.47) system

[SOURCE: ISO/IEC 7812-1:2017, 3.1]

3.2

Advanced Encryption Standard

AES

16-byte block *cipher* (3.3)

Note 1 to entry: This is defined in [ISO/IEC 18033-3](#).

3.3

algorithm

specified mathematical process for computation or set of rules which, if followed, will give a prescribed result

[SOURCE: ISO 16609:2022, 3.1]

3.4

archived key

inactive *cryptographic key* (3.28) that is being stored in a secure manner for a non-operational purpose

3.5

asymmetric algorithm

cryptographic algorithm (3.27) that uses two related keys, a *public key* (3.71) and a *private key* (3.69), where the two keys have the property that, given the public key, it is *computationally infeasible* (3.25) to derive the private key

3.6

asymmetric cryptosystem

cryptosystem using *asymmetric algorithms* (3.5)

3.7

authentication

provision of assurance that a claimed characteristic of an entity is correct

[SOURCE: ISO/IEC 27000:2018, 3.5]

3.8

authentication element

message element that is to be protected by *authentication* (3.7)

[SOURCE: ISO 16609:2022, 3.12, modified — Term revised.]

3.9

Base Derivation Key

BDK

key used in *derivation* (3.32) to generate *initial DUKPT keys* (3.45) for installation into transaction originating *secure cryptographic devices* (3.75) and for transaction processing

3.10

BDK ID

32-bit value that identifies the *Base Derivation Key* (3.9)

Note 1 to entry: This was formerly known as the *Key Set Identifier (KSI)* (3.57) in the TDEA DUKPT specification.

3.11

card acceptor

party accepting the card for the purpose of presenting transaction data to an *Acquirer* (3.1) or intermediary facilitating the transaction flow

[SOURCE: ISO/IEC 7812-1:2017, 3.3]

3.12

certificate

digitally signed statement that binds the value of a public key to the identity of the person, device or service that holds the corresponding private key

[SOURCE: ISO 20415:2019, 3.15, modified — Term revised.]

3.13

certificate authority

CA

entity that vouches for the binding between a device's identity, its public key and associated keying material

[SOURCE: ISO/IEC/IEEE 8802-11:2022, definition modified.]

3.14

certificate authority system

CA system

infrastructure required to manage, maintain and secure the key pairs and certificates of the *certificate authority* (3.13)

Note 1 to entry: A CA system will typically include one or more *Hardware Security Modules* (3.42), firmware, computer equipment, operating systems and software

3.15

certificate policy

CP

named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements

[SOURCE: ISO 21188:2018, 3.13]

3.16
certificate practice statement
CPS

statement of the practices which a *certificate authority* ([3.13](#)) employs in issuing *certificates* ([3.12](#)) and which defines the equipment, policies and procedures the certificate authority uses to satisfy the requirements specified in the *certificate policies* ([3.15](#)) that are supported by it

3.17
certificate subject
entity identified in a *certificate* ([3.12](#))

EXAMPLE *Secure cryptographic device* ([3.75](#)).

3.18
chain of custody
demonstrable possession, movement, handling and location of material from one point in time until another

[SOURCE: ISO/IEC 27050-1:2019, 3.1]

3.19
check value
key check value
KCV
component check value
CCV

non-secret value that is cryptographically related to the key (or component) and is used to verify that the underlying value is as expected

Note 1 to entry: It is possible for different keys or components to have the same check value.

3.20
cipher
method for the transformation of data in order to hide its information content, prevent its undetected modification and/or prevent its unauthorized use

3.21
ciphertext
data which has been transformed to hide its information content

[SOURCE: ISO/IEC 18033-1:2021, 3.7]

3.22
cleartext
plaintext
unencrypted information

[SOURCE: ISO/IEC 18033-1:2021, 3.20]

3.23
communicating pair
two parties (usually institutions) sending and receiving transactions

Note 1 to entry: This includes alternate processing sites either owned or contracted by either communicating party.

3.24
compromise
<cryptography> breach or failure of the security of a process or system used to protect the confidentiality or integrity of sensitive information

Note 1 to entry: Compromise includes situations in which either unauthorized disclosure of sensitive information could have occurred or appropriate control is not demonstrable.

3.25

computationally infeasible

property that a computation is theoretically achievable but is not feasible in terms of the time or resources required to perform it

3.26

credentials

identification data for an entity, incorporating at a minimum the entity's distinguished name and public key

3.27

cryptographic algorithm

algorithm (3.3) for the transforming of data using a *cryptographic key* (3.28)

Note 1 to entry: Data transformation includes operations such as *encryption* (3.40), *decryption* (3.31), synchronized generation of keying material and computation or verification of a digital signature or Message Authentication Code.

3.28

cryptographic key key

sequence of bits that determine the outcome of a cryptographic operation

Note 1 to entry: See cryptographic algorithm for further details.

Note 2 to entry: Examples of cryptographic operations include encryption, decryption, check function computation, derivation, signature generation or verification, and authentication.

3.29

cryptographic strength strength

computational cost of the most effective known attack against a given cryptographic algorithm and key size

Note 1 to entry: The cryptographic strength is usually a function of key size, but different algorithms can have different cryptographic strengths even though their key sizes are the same.

Note 2 to entry: The strength of a given algorithm and key size can vary depending on the use of the algorithm. For example, uses with few or no plaintext-ciphertext pairs could be more resistant to attack than uses with many pairs under a given key.

3.30

cryptoperiod

defined period of time during which a specific *cryptographic key* (3.28) is authorized for use or during which the cryptographic keys in a given system may remain in effect

[SOURCE: ISO 16609:2022, 3.9]

3.31

decryption

process of transforming *ciphertext* (3.21) data into *cleartext* (3.22) data

3.32

derivation

cryptographic transformation process that is used to derive one value from another value

Note 1 to entry: This is typically used to produce new *cryptographic keys* (3.28), which can potentially be used for different purposes, from a single key.

3.33

Derivation Identifier

Derivation ID

DID

32-bit value that identifies the specific *initial DUKPT key* (3.45) derived using the *Base Derivation Key* (3.9)

Note 1 to entry: This was formerly known as the 19-bit value Device ID (DID) or TRSM ID in the TDEA DUKPT specification.

3.34

derivation key

cryptographic key (3.28) that is used to cryptographically compute another key using *derivation* (3.32)

3.35

derived unique key per transaction

DUKPT

key management method that uses a unique key for each *transaction* (3.84) and prevents the disclosure of any past key used by the *transaction originating SCD* (3.86)

Note 1 to entry: The unique *transaction keys* (3.85) are derived from a *Base Derivation Key* (3.9) and non-secret data transmitted as part of each transaction.

3.36

digital certificate

asymmetric cryptosystem that provides for the creation and subsequent verification of *digital signatures* (3.37)

3.37

digital signature

cryptographic transformation of data which, when associated with a data unit and accompanied by the corresponding public-key certificate, provides the services of origin authentication, data integrity and signer non-repudiation

3.38

double-length TDEA key

TDEA key having a length of 128 bits, consisting of 112 key bits and 16 parity bits, which is typically represented in 32 hexadecimal digits

3.39

dual control

process of utilizing two or more separate individuals operating in concert to protect sensitive functions or information whereby no single individual is able to use the function or access all the information alone

Note 1 to entry: A *cryptographic key* (3.28) is an example of the type of material protected using dual control.

Note 2 to entry: For protecting cryptographic keys and other sensitive data, this concept is closely related to *split knowledge* (3.77).

[SOURCE: ISO 9564-1:2017, 3.10, modified — Definition revised and notes to entry added.]

3.40

encryption

encipherment

process of transforming *cleartext* (3.22) data into *ciphertext* (3.21) data for the purpose of security or privacy

3.41

exclusive-or

XOR

mathematical operation defined as: 0 XOR 0 = 0 0 XOR 1 = 1 1 XOR 0 = 1 1 XOR 1 = 0

Note 1 to entry: Equivalent to binary addition without carry (modulo-2 addition).

3.42 Hardware Security Module HSM

secure cryptographic device (3.75) that provides a set of secure cryptographic services including, but not limited to, key generation, cryptogram creation, PIN translation and certificate signing

[SOURCE: ISO 13491-1:2016, 3.23]

3.43 hash function

one-way function that maps a set of strings of arbitrary length on to a set of fixed-length strings of bits

Note 1 to entry: The output is generally relatively small.

Note 2 to entry: A collision-resistant hash function has the property that it is computationally infeasible to construct distinct inputs that map to the same output.

3.44 independent communication

process that allows an entity to counter-verify the correctness of a credential and identification documents prior to producing a certificate

EXAMPLE Call-back, visual identification.

3.45 initial DUKPT key IK

unique *cryptographic key* (3.28) loaded into a *secure cryptographic device* (3.75) that has been derived from a *Base Derivation Key* (3.9)

3.46 Initial Key ID

64-bit value that identifies the specific *initial DUKPT key* (3.45) derived under the *Base Derivation Key* (3.9)

Note 1 to entry: It is a concatenation of the *BDK ID* (3.10) and the *Derivation ID* (3.33)

3.47 Interchange

mutual acceptance and exchange of messages between institutions for card payment transactions

3.48 Issuer

institution holding the account identified by the Primary Account Number (PAN)

Note 1 to entry: See [Annex J](#) for additional information on the role an Issuer plays in the payment environment.

3.49 key agreement

process of establishing a shared secret key between entities in such a way that neither of them can predetermine the value of that key

Note 1 to entry: By predetermine, it is meant that neither entity A nor entity B can, in a computationally efficient way, choose a smaller key space and force the computed key in the protocol to fall into that key space.

[SOURCE: ISO/IEC 11770-4:2017, 3.13]

3.50 key component component

one of at least two values that are combined using XOR to form a symmetric *cryptographic key* (3.28)

Note 1 to entry: Each component has the same format (including length) as the cryptographic key.

3.51
key distribution host
KDH

host distributing keys in the *SKDAT* (3.79) key distribution methodology

Note 1 to entry: See [Annex K](#) for more details on the role a KDH plays in SKDAT.

3.52
key encryption key
KEK

key used exclusively to encrypt and decrypt other keys

3.53
key receiving device
KRD

device receiving keys in the *SKDAT* (3.79) key distribution methodology

3.54
key separation

method for ensuring that a key is used for only its intended purpose

3.55
Key Serial Number
KSN

concatenation of the *Initial Key ID* (3.46) and the transaction counter used in *DUKPT* (3.35)

3.56
key set

group of keys that are all determined by a common cryptographic procedure and differentiated by non-secret input such that knowledge of one key does not disclose any other key in the group

3.57
Key Set Identifier
KSI

non-secret value that uniquely identifies a *key set* (3.56)

3.58
key share

result of dividing a *cryptographic key* (3.28) into some number (n) of pieces (shares), such that a designated minimum number (m) of pieces are required to reconstitute the key

Note 1 to entry: Each key share is constructed in such a manner that access to fewer than m shares does not disclose any information about the key. In all cases, m is greater than 1 and less than or equal to n .

Note 2 to entry: A secret sharing scheme involving key shares is often referred to as an " m of n scheme".

3.59
keying material

data that comprise a complete *cryptographic key* (3.28) and its relevant metadata

EXAMPLE Keys, attributes, initialization vectors.

3.60
master key

highest level of key encrypting key in a hierarchy of *key encryption keys* (3.52) and *transaction keys* (3.85)

3.61
message authentication

verification that a message was sent by the purported originator to the intended recipient and that the message was not changed in transit

[SOURCE: ISO/IEC 20944-1:2013, 3.11.1.9]

3.62

Message Authentication Code

MAC

cryptographic value used to confirm that the message came from the stated sender and has not been changed in transit

3.63

node

point in a network that does some form of processing of data

EXAMPLE Terminal, Acquirer, switch.

3.64

non-repudiation

service that provides verifiable evidence to substantiate integrity and origin of data, thereby eliminating successful deniability

3.65

parity

result of a calculation of the number of '1' bits in a string of '0' and '1' bits that indicates whether the number of '1' bits is odd or even

3.66

payment instrument

physical payment card, electronic equivalent or other electronic instrument or order used for the transmission or payment of money, sold or issued to one or more persons, whether or not the instrument is negotiable

Note 1 to entry: This does not include any credit card voucher, any letter of credit or any instrument that is redeemable by the Issuer in goods or services.

3.67

Personal Identification Number

PIN

string of numeric digits established as a shared secret between the account owner and the *Issuer* ([3.48](#)), for subsequent use to validate authorized card usage

[SOURCE: ISO 9564-1:2017, 3.19, modified]

3.68

Primary Account Number

PAN

assigned number, composed of an Issuer identification number, an individual account identification and an accompanying check digit that identifies the card Issuer and account owner (who is the cardholder for physical cards)

Note 1 to entry: The accompanying check digit is specified in [ISO/IEC 7812-1](#).

[SOURCE: ISO 9564-1:2017, 3.22, modified]

3.69

private key

key in an asymmetric key pair which is known only by that entity

3.70

pseudo-random process

process that produces a value which is statistically random and essentially unpredictable even though it was generated by a deterministic algorithm

3.71

public key

key of an entity's asymmetric key pair which can usually be made public without compromising security

[SOURCE: ISO/IEC 11770-1:2010, 2.36]

3.72

random

value in a set that has an equal probability of being selected from the total population of possibilities, hence is unpredictable

3.73

recipient

person, institution or other party that is responsible for and authorized to receive a message or package

3.74

replay

process of sending a message which contains all or part of a previously sent message, as a method of perpetrating a fraud

3.75

secure cryptographic device

SCD

device that provides physically and logically protected cryptographic services and storage and which can be integrated into a larger system such as an ATM or POS terminal

EXAMPLE entry device (PED), Hardware Security Module (HSM).

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

3.76

sender

person, institution or other entity transmitting a message or package

3.77

split knowledge

condition under which two or more individuals separately and confidentially have information that, individually, convey no knowledge of the resulting combined information

EXAMPLE *Components (3.50) of a cryptographic key (3.28) managed under strict custody separation such that no one person can gain access to the entire set.*

[SOURCE: ISO 9564-1:2017, 3.28, modified — Definition revised and example added.]

3.78

symmetric key

secret *cryptographic key* (3.28) that is used in a symmetric *algorithm* (3.3)

3.79

symmetric key distribution using asymmetric techniques

SKDAT

key distribution method that uses asymmetric cryptography to protect the keys during transport

Note 1 to entry: See [Annex K](#).

3.80

switch

node that can route data from a node to other nodes

3.81

tamper evident and authenticable bag

TEA bag

one-time use packaging that is designed in such a way as to make it infeasible to access the contents without detection and has a pre-printed unique identifier that cannot be changed without detection to allow for authentication

Note 1 to entry: The form of the packaging is not necessarily restricted to a bag.

3.82

tampering

unauthorized modification that compromises the security properties of a device or security container

EXAMPLE 1 Insertion of active or passive tapping mechanisms into a device to determine, record or modify secret data.

EXAMPLE 2 An attempt to penetrate or open a security container, TEA bag or device.

3.83

terminal

device or system that initiates a *transaction* (3.84) and is, contains or interfaces with a *transaction-originating SCD* (3.86) for cryptographic functions

3.84

transaction

series of messages to perform a predefined function

EXAMPLE Payment transaction.

3.85

transaction key

key used to cryptographically protect the transaction data elements between nodes

Note 1 to entry: If more than one key is used for different cryptographic functions, each key can be a variant or derivative of the transaction key.

Note 2 to entry: A transaction key is sometimes referred to as a data key, communications key, session key or working key.

3.86

transaction-originating SCD

secure cryptographic device (3.75) that is, is integrated in or is used by the *terminal* (3.83) that initiates the financial *transaction* (3.84)

EXAMPLE PIN entry device.

Note 1 to entry: An intermediate device that processes the transaction, such as a *Hardware Security Module* (3.42), is not a transaction-originating SCD.

3.87

Triple Data Encryption Algorithm

TDEA

symmetric *algorithm* (3.3)

Note 1 to entry: The TDEA is specified in the [ISO/IEC 18033](#) series.

3.88

triple-length TDEA key

Triple Data Encryption Algorithm (3.87) key having a length of 192 bits, consisting of 168 key bits and 24 parity bits, that is typically represented in 48 hexadecimal digits

3.89 verification

process of associating and/or checking a unique characteristic

4 Key management requirements

4.1 General

4.1.1 Key management strategy

Key generation, storage, loading, transport, use and destruction are points at which a key management policy is applied to ensure the secrecy of the keys. For an illustration of the symmetric key life cycle, see [Annex D](#); for the asymmetric key life cycle, see [Annex E](#).

The impact of a compromise of a given key should be taken into consideration when deciding the policy for management.

It is necessary to employ mechanisms that detect or prevent the unauthorized use of a key, the use of modified or substituted keys or the replay of keys.

This subclause includes general requirements that apply to the management of keys.

4.1.2 Dual control and split knowledge of secret or private keys

The management of cleartext components or shares of symmetric (secret) and asymmetric private keys shall ensure dual control and split knowledge of the resulting key. Throughout the life cycle of a key, an individual shall not have (or have had) access to all cleartext components or a sufficient number of shares to form the key. Each person or group of persons (e.g. custodian) responsible for each key component or share shall have received adequate training and shall have signed an acknowledgement of their responsibilities to keep secret the key component or share entrusted to them.

The chain of custody for each cleartext component or share shall be documented and such documentation shall be maintained.

Access to fewer than the number of shares required to reconstruct the plain text symmetric or private key or components required to reconstruct a symmetric key shall give no information about the overall key.

A cleartext key share or component of a symmetric or private key shall be accessible only to the person or group of persons to whom it has been entrusted and only for the minimum duration required.

4.1.3 Permissible key forms

4.1.3.1 Permissible symmetric key forms

Symmetric cryptographic keys shall only exist in one or more of the following forms:

- a) in a secure cryptographic device (SCD) as specified in [4.2](#);
- b) if encrypted and outside an SCD:
 - 1) as a cryptogram of the key that has been created inside an SCD using an AES or TDEA key encrypting key (see [4.1.5](#) for applicable key strength and [4.5](#) for key block requirements);
 - 2) as cryptograms of key shares or key components, where the cryptograms have been created inside an SCD using an AES or TDEA key encrypting key (see [4.1.5](#) for applicable key strength and [4.5](#) for key block requirements);

- 3) as cryptograms of the key, key shares or key components that have been created inside an SCD using an asymmetric algorithm listed in [Annex F](#).
- c) if not encrypted and outside of an SCD:
 - 1) as two or more full-length key components that are managed in accordance with [4.1.2](#), thereby achieving dual control and split knowledge of the key;
 - 2) as two or more key shares that are created using a secret sharing method which is perfectly information-theoretically confidential, is defined in [ISO/IEC 19592-2](#) and managed in accordance with [4.1.2](#), thereby achieving dual control and split knowledge of the key.

A component shall only be associated with a single key, except by chance.

A key encrypting key that protects a large number of keys, such as an Acquirer or Issuer top-level key that is managed as components or shares, should be managed as at least three separately managed components or shares, so that it requires at least three people to load the key.

4.1.3.2 Permissible asymmetric key forms

4.1.3.2.1 Asymmetric private key forms

Asymmetric private keys shall only exist in one or more of the following forms:

- a) within an SCD as specified in [4.2](#).
- b) if encrypted and outside an SCD (including transport between SCDs):
 - 1) as a cryptogram of the key that has been created inside an SCD using an AES or TDEA key encrypting key (see [4.1.5](#) for applicable cryptographic strength and [4.5](#) for key block requirements);
 - 2) as cryptograms of key shares, where the cryptograms have been created inside an SCD using an AES or TDEA key encrypting key (see [4.1.5](#) for applicable key strength and [4.5](#) for key block requirements);
 - 3) as cryptograms of the key or key shares that have been created inside an SCD using RSA OAEP or ECIES as defined in [ISO/IEC 19592-2](#);
- c) if not encrypted and outside of an SCD, only as two or more key shares that are created using a secret sharing method which is perfectly information-theoretically confidential, is included in [ISO/IEC 19592-2](#) and managed in accordance with [4.1.2](#), thereby achieving dual control and split knowledge of the key.

4.1.3.2.2 Asymmetric public key forms

In an asymmetric cryptosystem there is no secrecy requirement for the storage of the public key, but authenticity and integrity of this key shall be ensured before use, preventing substitution or alteration of the public key or associated information without detection.

One or more of the following techniques shall be used to ensure public key integrity:

- sign the public key and associated data using a digital signature system, thereby creating a public key certificate in accordance with [4.3](#) and [4.4](#);
- create a MAC for the public key and associated data, using an algorithm defined by [ISO 16609](#) and a key used only for this purpose;
- depend on protections of an SCD (see the [ISO 13491](#) series);
- use the key verification techniques as defined in [4.8.3](#) (when the public key is stored in plain text).

4.1.4 Logging

4.1.4.1 General

All instances in which cleartext key components or shares are created, received, accessed, transported, loaded, stored or destroyed shall be recorded. A policy that defines the length of time such records are maintained shall exist and be in use. Consideration should be given to the possible need to inspect those records for a finite length of time after the key is removed from service. At a minimum, records kept by each party managing the key shall reflect the date, time, person(s) involved, activity type (e.g. generation, destruction, shipment), reason for activity, SCD Identifier (if applicable) and TEA bag number(s) (if applicable).

Except in the case of intermediate or ephemeral keys that exist only within a specific SCD and only for the length of time necessary to execute a cryptographic function, instances where cryptographic keys are created, transported, received, replaced or destroyed without manual intervention (e.g. dynamic key exchange) should be logged in such a way as to provide evidence that the event occurred. Similarly, instances where cryptograms are created, received, transported, loaded or stored with manual intervention should be logged in such a way as to provide evidence that the event occurred. Such logging may be done outside the SCD.

4.1.4.2 Asymmetric keys

All operations involving private keys shall be recorded.

The integrity of the logs shall be preserved.

For CA operations involving private keys, logs shall include reference to all hardware used in support of these activities. Hardware reference shall include SCD serial numbers and a means to distinguish make and model as a minimum. Other device attributes may also be recorded. The logs may be manual and/or electronic and the information may be contained in a combination of logs.

Operations involving the public key shall be recorded as indicated in [Table 1](#).

Table 1 — Public key recording matrix

	Created	Signed	Installed as trusted key	KDH bind	KDH unbind	Destroyed
CA	✓	✓	✓			✓
KDH	✓	✓	✓	✓	✓	✓
KRD	✓	✓	✓			
NOTE Events related to installation or removal of a KDH public key in KDH bind and KDH unbind are typically logged at the KDH rather than at the KRD, even though the KRD is the device that is receiving or removing the KDH public key.						

4.1.4.3 SCD hardware logging

Chain of custody logging shall be maintained for all SCDs, excluding access related to the use of deployed transaction-originating SCDs.

Approved settings for user-defined parameters and configurations for HSMs shall be defined and logs reflecting those chosen during configuration activities shall be maintained.

4.1.5 Cryptographic strength

Cryptographic strength of keys shall be at least 112 bits.

The strength of a key that is encrypting other keys should be equal to or greater than the key being encrypted. A key ever encrypted by a weaker key inherits the cryptographic strength of the weakest key with which it was ever encrypted. Observe that this weakening propagates down through a key

hierarchy. For example, encrypting an AES 128 key with a triple-length TDEA key reduces the AES key's cryptographic key strength to that of the TDEA key, which is 112 bits. See [ISO/TR 14742](#) for more details on cryptographic strength where various algorithms are used. A means to track the effective cryptographic strength of a key shall exist.

If TDEA is used for a new implementation, such implementation should utilize unique keys per transaction.

4.1.6 Key locations

A symmetric or private key shall exist in the minimum number of locations as is necessary for operations. The fewer the instances of the key, the stronger the claim of non-repudiation, as there is a lower probability of compromise.

4.1.7 Single-purpose key usage

4.1.7.1 Single-purpose symmetric key usage

An HSM MFK shall not be used as a key between organizations or logical systems for transport of other keys. A logical system would include a single system with multiple HSMs

Other than HSM master keys, each symmetric key shall have an intended purpose designation (IPD). Such IPDs shall be used by the SCD to restrict the use of the key to no more than one item from each of the following four classifications:

- a) algorithm with which the key is to be used:
 - 1) AES;
 - 2) TDEA;
 - 3) HMAC;
- b) cryptographic function:
 - 1) encryption/decryption;
 - 2) authenticated encryption;
 - 3) derivation;
 - 4) message authentication;
 - 5) key wrapping;
- c) data element or type of information involved:
 - 1) account holder authentication element (e.g. PIN) ;
 - 2) keys;
 - 3) other data element not listed previously;
- d) functional direction or mode of use:
 - 1) encrypt/generate/wrap;
 - 2) decrypt/verify/unwrap;
 - 3) both.

Where the IPD is used to indicate functional direction, the same key may have more than one IPD, depending on its environment. IPD configurations enforceable by the SCD shall be cryptographically bound to the key.

Keys electronically transmitted from the generating SCD shall have the IPD identified and bound to the key, which can be accomplished by setting attributes in a key block.

Keys shall be given the most restrictive IPD consistent with the function the key will be supporting in the implementation (e.g. encryption only or decryption only as opposed to both encryption and decryption). There are keys that are by nature bidirectional, such as a master key within an SCD (e.g. master file key).

4.1.7.2 Single-purpose asymmetric key usage

Each asymmetric key pair shall have an IPD. Such IPDs shall be used by the SCD to restrict the use of the key to only one item from each of the three following classifications:

- a) algorithm with which the key is used;
- b) cryptographic function (except for self-signed CSR):
 - 1) encryption;
 - 2) decryption;
 - 3) signing;
 - 4) verifying;
 - 5) key agreement;
- c) Data element or type of information involved, including:
 - 1) keys related to confirming SCD legitimacy and/or functionality (e.g. manufacturer keys for verification of legitimate parts, code)
 - 2) keys related to SKDAT;
 - 3) PINs;
 - 4) data.

The IPD should be used by the SCD to restrict the use of the key for communicating pairs to either:

- a) terminal-to-host; or
- b) host-to-host.

Each key shall be given a single IPD. IPD configurations enforceable by the SCD shall be cryptographically bound to the key

Keys electronically transmitted from the generating SCD shall have the IPD identified and bound to the key, which can be accomplished by setting attributes in a key block.

4.1.7.3 Additional intended purpose designations (AIPD)

In addition to the segmentation required previously, AIPDs may also be used to provide for segmentation classifications that are enforced by policy and procedure, rather than by the SCD.

An AIPD should be used to restrict the use of the key for communicating pairs to either:

- a) terminal-to-host; or

b) host-to-host.

Examples of other AIPD classifications include, but are not limited to:

- designation as production or non-production;
- additional data element segmentation (e.g. separating PAN from other data types or segmenting an implementation-specific data element);
- BDK segmentation; for entities processing or injecting DUKPT keys or keys for other key-derivation methodologies, this might be done to segment the risk associated with a BDK compromise.

4.2 Secure cryptographic device

4.2.1 General requirements

Keys are managed within devices such as PIN entry devices, Hardware Security Modules and key loading devices. These devices shall meet the requirements of an SCD as described in the [ISO 13491](#) series.

Some key management activities necessitate the use of non-SCD devices where their form factors have inherent limitations preventing them from meeting some SCD security requirements, especially tamper-responsiveness. These limitations have associated security risks which shall be addressed by restricted usage and additional controls. These devices are hardware management devices (HMD). Examples of HMDs include, but are not limited to:

- a) smart cards used for component or share transport or storage;
- b) smart cards containing public or private key pair(s) used to facilitate management of HSMs;
- c) devices used to authorize or enable key management functions.

HMDs shall only be used in cases where the compromise of a single HMD would not compromise keys or secrets not held within that HMD.

Where there are requirements in this document that apply to an SCD, such requirements shall also apply to an HMD, unless specifically excluded.

If multiple cleartext components or shares sufficient to form a key are stored or transported within a single SCD, it is equivalent to having a clear key in the SCD; therefore, the SCD shall meet tamper responsive requirements in [ISO 13491-1](#). The SCD shall enforce management of the components or shares in such a way as to ensure split knowledge of the key in accordance with [4.1.2](#).

When an SCD is used to generate a key or key component, the SCD shall use a random or pseudo-random process such that certain values are not statistically more probable than others from the set of all possible values, in accordance with [ISO/IEC 18031](#).

When a key is entered as components, the SCD shall combine all the components using an XOR operation. [Figure 1](#) is an example of the XOR process used to combine key components.

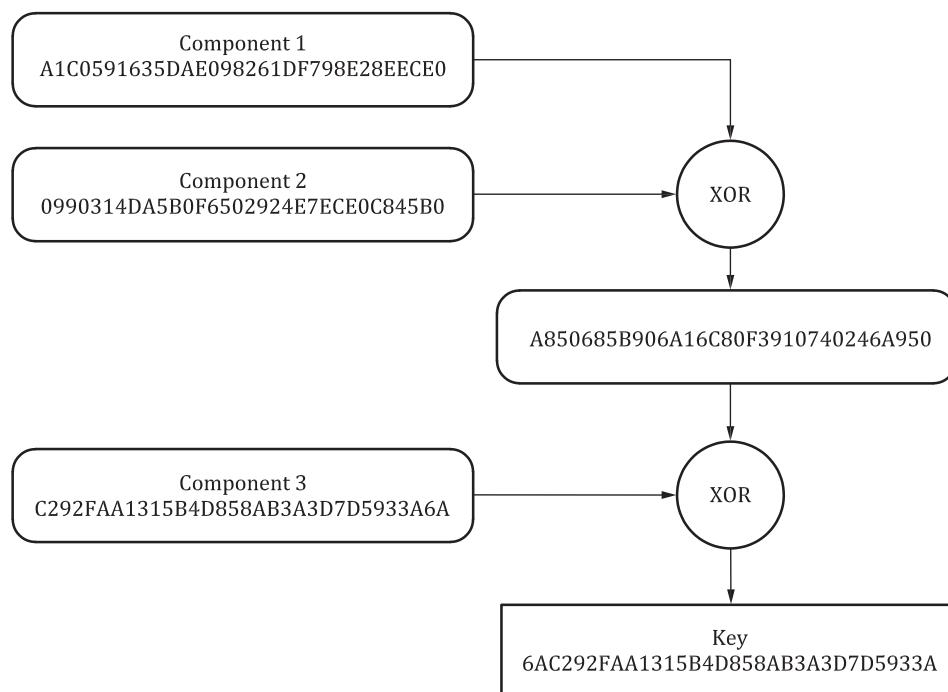


Figure 1 — Example of XOR function to combine key components

When an SCD is used to produce key shares, the SCD shall use an m -of- n secret-sharing scheme which is perfectly information-theoretically confidential and defined in [ISO/IEC 19592-2](#).

The management and control of an SCD shall conform to the [ISO 13491](#) series.

An HMD or an SCD that does not have a tamper responsive mechanism enabled shall be stored in an environment that meets at least the criteria of a minimally controlled environment as identified in the [ISO 13491](#) series, with additional controls providing reasonable assurance that any unauthorized access to the device can be detected. Because the device cannot respond to an attack, the additional controls minimize the probability of unauthorized modifications.

An SCD is decommissioned when it is removed from service for repair, ownership is transferred to another organization or the SCD is to be destroyed. When an SCD is decommissioned, the financial keys shall be erased in accordance with [ISO 13491-1](#). The keys should be erased as close to the time of removal from service as practical.

When an SCD is lost or stolen, all symmetric or private keys contained in that device should be considered compromised and the investigation outlined in [4.17](#) shall be initiated.

4.2.2 Additional SCD requirements for devices used in SKDAT

A KRD shall have mechanisms to prevent or detect middleperson attacks during the execution of SKDAT. A KRD shall have mechanisms that provide secure processing for authentication of the interfacing KDH.

A KDH shall have mechanisms to prevent or detect middleperson attacks during the execution of SKDAT and the ability to ensure communication with only an intended KRD. A KDH shall have mechanisms that provide secure processing for mutual authentication of the interfacing KRD.

A KRD may interface with more than one KDH for different key types (e.g. one for PIN keys and one for PAN encryption keys offered by a point-to-point encryption solution provider).

The first time a KRD interfaces with a KDH for a given key type (e.g. delivery of PIN-related keys), such mechanisms might only be sufficient to identify the KDH as having been one of a group of KDHs authorized by the device manufacturer or CA for that purpose. Once the KRD is authenticated to

the intended KDH for a given key type, mechanisms shall exist to prevent communication with any unintended KDH for that same purpose.

4.3 Additional CA requirements

Where a CA is used for purposes of verifying the identity of participants in a key exchange protocol, then for certificate issuance the CA shall have processes in place to validate the identity of the certificate requester before issuing a digital certificate for the requester's associated public key.

The CA shall be implemented in accordance with the requirements specified in [ISO 21188](#).

The CA shall document a certificate policy (CP) as specified in RFC 3647.

The CA shall document a certification practice statement (CPS) in accordance with RFC 3647 and the CA shall operate in accordance with its CPS.

In the event of a compromise of the CA system's cryptographic keys, the CA shall have procedures in place to revoke certificates and notify affected parties.

The CA system shall operate in an environment that meets the criteria of a secure environment as described in the [ISO 13491](#) series, which provides secure processing to ensure the integrity of the public key infrastructure environment.

The CA system root private key shall only be operated to issue, revoke and suspend subordinate CA certificates. The CA system root private key shall be operated in a secure environment as defined in the [ISO 13491](#) series. The CA system root private key shall be operated offline, i.e. in an "air-gapped" closed system. Access to the CA system root private key shall only be under controlled conditions using principles of dual control, observed by an independent witness. Audit or procedure records of each CA system root private key access shall be maintained such that all activities performed can be identified from the records alone.

All CA functions, including any registration functions, shall be executed on hardware that is physically separate from the key distribution host hardware.

4.4 Additional RA requirements

The Registration Authority (RA) is the function that is responsible for identification and authentication of the certificate subject (the entity identified in the certificate), but as it does not perform CA functions, it does not sign or issue certificates. An RA might assist in the certificate application process or revocation process or both. The RA does not need to be a separate body, but may be part of the CA.

The CA shall verify or require that the RA verifies the credentials presented by a certificate subject as evidence of identity or authority to perform a specific role in accordance with the certificate policy. The RA shall verify the certificate subject's possession of the associated private key through the use of a digitally signed certificate request pursuant to PKCS #10 or another cryptographically equivalent demonstration.

The RA shall also perform validation that the requestor is authorized by doing at least one of the following:

- confirmation by telephone, confirmatory postal mail and/or comparable procedure to the certificate subject to confirm that the organization has authorized the certificate application, confirmation of the employment of the representative submitting the certificate application on behalf of the certificate applicant and confirmation of the authority of the representative to act on behalf of the certificate applicant;
- confirmation by telephone, confirmatory postal mail and/or comparable procedure to the certificate subject's representative to confirm that the person named as representative has submitted the certificate application;

- confirmation that the requestor is authorized based on previously established trusted communication between the requestor and the RA (e.g. the SCD is directly connected to the certificate issuing CA within the manufacturer or repair zone or sending the public key of a KDH to the CA for signing over an authenticated communication channel).

The RA registration process to validate the certificate subject (SCD) may include, but is not limited to:

- checking the certificate subject's credentials evidencing claimed identity;
- verifying that identifying data provided by the certificate subject is valid;
- verifying that the identifying data pertain to the certificate subject;
- verifying that the certificate subject is entitled to obtain a public key certificate under the certificate policy;
- recording the binding of the certificate subject's identification data to the certificate subject's public key;
- checking the certificate request for errors or omissions in accordance with the certificate policy;
- direct inspection of the SCD;
- verifying a previously installed certificate;
- verifying that the serial number is on a validated list.

4.5 Key blocks

4.5.1 Overview of key blocks

When in storage or transmission, symmetric and private asymmetric keys shall be protected against misuse, substitution and modification through the use of key blocks. Public keys may be stored in the same general format of a key block without using encryption. This requirement applies to all symmetric algorithms and key sizes.

The key block protects the key and confidential attributes from unauthorized disclosure (using encryption) and protects all data in the key block from modification (using integrity protection).

[Figure 2](#) is a high-level illustration of the structure a key block may have but is not intended to be prescriptive.

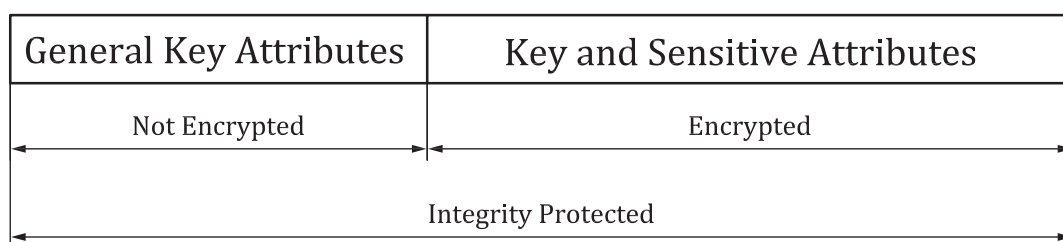


Figure 2 — Key block overview example

Except in the case of public keys, the key and its confidential attributes shall be encrypted. If the secrecy of the attribute is important to security of the key and key block, it shall be considered a confidential attribute.

The key and its attributes in the key block shall have integrity protection such that they cannot be modified without detection.

The key block shall contain the entire encrypted key (or component or share) and any attributes that are required according to this document. In addition, either the encrypted or the unencrypted portion of the key block may have additional data that is outside the scope of this document.

The process to create the key block is "key wrapping". [ISO 20038](#) is an example of using an AES key to create a key block that meets the requirements in this subclause. ANSI X9.143 is an example of using a TDEA key to create a key block that meets the requirements in this subclause.

4.5.2 Key attributes

The key attributes define how the key can be used and other pertinent information. These are represented by values known as key tags, which are distinct constants (e.g. as defined in [ISO 20038](#) for AES and ANSI X9.143 for TDEA). At a minimum, the attributes shall include the following:

- a) One or more attributes as set by the IPD that define the operations for which the key can be used. Examples of attributes that define permitted operations include ones that specify that the key can be used only for PIN block encryption, only for encrypting other keys or only for computing Message Authentication Codes (MACs). Other attributes can define sub-categories, for example specifying whether a MAC key can be used to generate a MAC, to verify a MAC or both.
- b) One or more attributes that define the cryptographic algorithm and mode for which the key can be used. These attributes are intended to prevent the misuse of a key using a different cryptographic algorithm or mode that could facilitate an attack to determine the value of the key.

Security requirements for key attributes are specified in [4.5.1](#).

4.5.3 Integrity of the key block

Before the key in the key block is used, the SCD that will use it shall verify the integrity of the key block with a cryptographic method that uses a key which is at least equal in strength to the key being protected. The key in the key block shall only be used if the integrity check passes.

4.5.4 Key and sensitive attributes field

4.5.4.1 Content of the field

At a minimum, the key and sensitive attributes field of the key block shall contain the value of the key but may also contain other data. For example:

- a) Padding that has been used to extend the length of the cleartext key to some desired length before it is encrypted. One purpose of this is to pad a key to the block size of the algorithm used to encrypt it. Another purpose is to hide the true length of the cleartext key, for example by padding so that all keys have the same encrypted length.
- b) Information indicating the length and location of the key to assist in parsing.
- c) Any sensitive attributes of that key that need to be kept secret.

4.5.4.2 Encryption of the key and sensitive attributes field

Electronic Code Book mode shall not be used to encrypt this field if the field is longer than one block.

The key and sensitive attributes field encrypted with AES shall use one of the methods identified in [ISO 20038](#) or [ISO/IEC 19772](#).

Methods using TDEA keys to encrypt the key and sensitive attributes field include, but are not limited to:

- the TDKW method from ANSI X9.102;
- methods defined in ANSI X9.143.

For further guidance, see [ISO/TR 14742](#).

4.6 Key creation

4.6.1 Symmetric key creation

4.6.1.1 Random key generation

Keys shall be generated by using a random or pseudo-random process that ensures that no key generated is more probable than any other key. The strength of the random number generator shall be equal to or greater than the strength of the key it is generating. For further guidance on random number generation, see [ISO/IEC 18031](#).

Keys shall be generated within an SCD.

4.6.1.2 Key derivation

Key derivation is a technique by which a (potentially large) number of keys are created (“derived”) from a single secret key and non-secret variable data. The original single key is called a “derivation key” or “base key” and each key generated from it is called a “derived key”. Key derivation can be used with a variety of key management methodologies and algorithms. For example, DUKPT allows for the derivation of a large number of statistically unique keys from a single key, eliminating the need to store each subsequent derived key. This is useful for an entity that drives a large number of devices, where each device is required to have unique keys.

Unique key per transaction (UKPT) is an example of a key derivation method used between two hosts. UKPT is a key management method to generate a session key based on a base key used by two communication partners exchanging a random value or counter to derive a unique key valid for the transmission of exactly one message or message exchange. It is up to the two organizations involved to determine the information that will be used to derive the unique key(s) for the transaction. See [ISO/IEC 11770-6](#) for guidance.

It shall be infeasible to recover the derivation key from any set of derived keys. The procedure may be used iteratively, as a key generated from one initial key may subsequently be used as a derivation key to generate others. The derivation function shall conform to [ISO/IEC 11770-6](#).

4.6.1.3 Key calculation (variants)

Applying a variant is a reversible technique for obtaining a set of keys from a single key, with each resulting key having a different key type. This technique provides key separation while eliminating the need to manage a separate, unrelated key of each required type. Each variant key is calculated from the original key and one constant from a set of non-secret constants using a reversible process. An example of such a process is the modulo-2 addition (i.e. XOR) of the key and a non-secret value. The disclosure of one key calculated in this manner discloses the original key and all other keys calculated from it.

A variant of a key shall be used only for key separation based on usage and the variant(s) of a key shall only exist in the same SCD that contains the original key.

Where variants are used with double-length TDEA keys, and one variant key is used to encrypt data that will also be available in the clear (e.g. PANs), the original key and associated variant keys shall be replaced regularly, in accordance with [4.13](#).

If a key is compromised, then all variants of the key are also compromised. If there are multiple variants of a key, and any one of those variant keys is compromised, then all variants and the original key are also compromised.

4.6.2 Asymmetric key creation

4.6.2.1 Introduction

The asymmetric key pair generation is the process by which a new pair of keys composed of a private key and the related public key is generated for use in a specific asymmetric cryptosystem. The two keys of an asymmetric key pair are mathematically related as defined by the design of the particular asymmetric cryptosystem. The relationship is such that it is computationally infeasible to determine the private key from the public key.

4.6.2.2 Requirements

The following requirements in this subclause apply regardless of the organization performing the generation activities.

Asymmetric key pairs shall be generated:

- a) only within an SCD as specified in [4.2](#);
- b) in such a way that guarantees the secrecy of the private key and the integrity of the public key;
- c) in accordance with relevant reference(s) (the [ISO/IEC 11770](#) series for RSA and Diffie-Hellman, ANSI X9.63 for Elliptic Curve and [ISO/IEC 18032](#) for Prime Number Generator);
- d) using an ISO-approved method for random number generation (see [ISO/IEC 18031](#)).

If the key pair is generated by a system that will not use it, the following applies to the transfer or transport of the key pair:

- The private key shall not be available in human-comprehensible form, i.e. it shall be encrypted.
- The private key of the key pair and all related secret seed elements shall be deleted from the sending SCD immediately after the transfer has been confirmed.
- The integrity and confidentiality of the private key shall be ensured throughout the transfer process.
- The public key shall only exist in one of the forms in [4.1.3.2.2](#) and should be delivered within a certificate if a CA is used.

Unique asymmetric key pairs shall be generated for each function (e.g. encryption versus signature) for each entity.

Asymmetric key pairs should have a replacement date to establish their life cycle.

4.7 Key component and key share creation

Both symmetric and asymmetric keys can exist as an m of n key share scheme with a minimum of two shares. Only a symmetric key can exist as two or more full-length components. The processes to create components or shares shall be performed using the principles of dual control and split knowledge. Components can be generated individually and combined to form a key or created from an existing key. The process to manage a key as shares requires the generation of the key and subsequent creation of the shares.

Key components shall be created by using a random or pseudo-random process such that no value is statistically more probable than any other from the set of all possible values. See the [ISO/IEC 18033](#) series.

Key shares shall be created using a secret sharing method defined in [ISO/IEC 19592-2](#), which is identified as perfectly information-theoretically confidential (also known as perfectly information-theoretically secure).

Except by chance, an individual key component shall only be used in the formation of a single key (i.e. the reuse of a component to form more than one key is not permitted).

Key shares and components for a new or existing key shall be created within an SCD.

If component creation or fragmentation into shares produces clear, human readable values, the process shall occur in at least a controlled environment as defined in the [ISO 13491](#) series.

During the creation process, cleartext components shall only be displayed on or printed by an SCD. Cleartext components produced during creation and displayed on the SCD may be recorded on paper by the assigned key custodian.

If components are to be stored or transported, TEA bags shall be used. TEA bags are not required if components are created, loaded and destroyed in the same activity and location.

4.8 Check values

4.8.1 Introduction

A check value (CV) is used to verify that a value (e.g. a key, key component or key share) is as expected without disclosing the value itself. At any time following initial generation of the CV, repeating the same function with the same value will produce the same results. If the resulting CV is identical to the initial CV, it is assumed that the value is unchanged.

CV verification can be used to reasonably establish that one or more of the following conditions have been met:

- A value has been correctly entered into a cryptographic device.
- A value has been correctly received over a communications channel.
- A value has not been altered or substituted.

For example, an organization receiving a key, component or share from another organization might wish to have a method to confirm that the values received have been loaded correctly. CVs generated on keys are called key check values (KCVs). CVs generated on key components are called component check values (CCVs).

NOTE Because of the limited size of the CV, in a large set of cleartext values (keys, components, shares), different values can have the same CV (collisions).

If a KCV is used to verify keys and the calculated KCV does not match the expected KCV, the key shall not be used until an investigation is performed to determine the cause of the mismatch.

4.8.2 Symmetric key check value calculation

A check value should be calculated with a key or component at the time of generation. When a check value is calculated for a key or key component, it shall be calculated by a cryptographic process such that all portions of the key or key component are involved in calculating the check value. It is a significant security risk to calculate check values on portions of the key individually (e.g. a separate check value for the right and left halves of a double-length TDEA key). Thus, in the description of this algorithm, the term “key” always refers to the entire key (e.g. all 112 bits of a double-length TDEA key and all 192 bits of a triple-length TDEA key).

[Annex A](#) details symmetric key check value calculation. When calculating CVs for AES keys, [A.3](#) shall be used. When calculating CVs for TDEA keys, [A.2](#) or [A.3](#) shall be used. When generating a key check value for an HMAC key, it shall be computed according to [A.4](#).

When displaying or recording the component check value or key check value computed according to [A.2](#), no more than the six hexadecimal digits of the cipher text shall be used.

When displaying or recording the component check value or key check value computed according to [A.3](#), no more than the ten hexadecimal digits of the cipher text shall be used.

4.8.3 Asymmetric key check value calculation

For asymmetric keys, the reference KCV may be computed as the hash of the public key, and optionally associated data, using an algorithm defined in [ISO/IEC 10118](#).

For public keys, as long as the KCV is distributed via an integrity-assured channel, the public key can be distributed via a non-secure channel. Prior to installing the public key for use, the user shall validate it by recomputing the KCV and comparing it with the reference KCV.

It shall be infeasible to modify or substitute the reference KCV and public key (and associated data) in such a way that the recomputed KCV of the modified or substituted public key (and associated data) equals the modified or substituted reference KCV. This can be achieved by either of the following:

- ensuring the integrity of the KCV (see [ISO 16609](#));
- separately storing or distributing the reference KCV in such a way that modification or substitution of the reference KCV cannot be coordinated with modification or substitution of the public key (and associated data) (e.g. dual controls).

4.9 Key distribution

4.9.1 Symmetric key distribution

4.9.1.1 Introduction

The distribution process for an encrypted key shall protect against the use of a key that has been substituted or modified in an unauthorized manner.

Compliant implementation of requirements for secure management of symmetric keys requires (among other things) unique key relationships and strict enforcement of dual control and split knowledge processes when handling cleartext keying material. This includes keys deployed to remote devices or established between communicating pairs. Historically, compliant implementation of key distribution has been a manually performed, physically on-site process that is difficult to manage, costly and/or non-existent (i.e. not compliant). An automated rather than manual method of distributing symmetric keys can address these issues and result in improved security of the financial services environment.

4.9.1.2 Encrypted with symmetric key

Keys, components, and shares encrypted with symmetric keys and contained in key blocks in accordance with [4.1.3.1](#) can be transmitted through unprotected communication channels.

4.9.1.3 Encrypted with asymmetric key

4.9.1.3.1 General

The use of public key cryptography and associated asymmetric key algorithms is one solution for automated remote symmetric key distribution, and therefore included in this document. This solution is referred to as symmetric key distribution using asymmetric techniques (SKDAT).

The functional roles for SKDAT are described in [Annex K](#).

For SKDAT, two types of symmetric keys can be involved. The first is the symmetric key (e.g. key encrypting key, PIN encrypting key, DUKPT initial key) to be installed into the KRD and corresponding Acquirer host (if applicable) or to be distributed between two hosts. The second is a symmetric key that functions as a key encrypting key for purposes of encrypting a private key of a public-private key pair for storage in a database or during transport. The first type is the object of protection of this document.

in 4.9.1.2; the second provides the protection and is subject to all the requirements of other symmetric keys. A SKDAT implementation shall include a method to bind the symmetric key being distributed to a particular cryptographic algorithm.

For automated SKDAT, examples of acceptable methods of binding the symmetric algorithm to the distributed key include but are not limited to:

- using cryptographically bound mechanisms, for example labels, to indicate the appropriate algorithm and usage;
- including the symmetric algorithm name or other unique identifier in the key derivation function of a key agreement scheme.

Keys, components and shares encrypted with asymmetric keys as described in 4.1.3.1 can be transmitted through unprotected communication channels.

At a high level, there are three phases of SKDAT in typical protocols using a CA. The following description is based on a PIN entry device (PED) as the KRD, but the concepts can apply to any KRD.

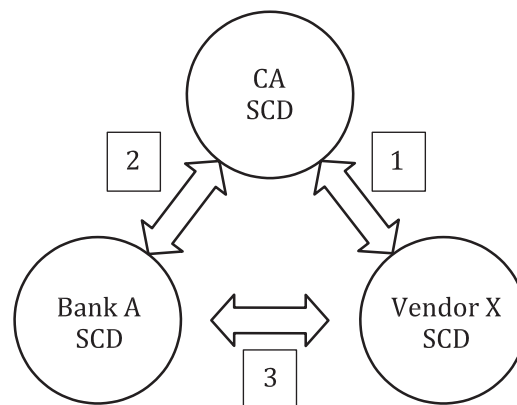


Figure 3 — Automated SKDAT phases

Phase 1 – KRD initialization: During the manufacturing phase of the KRD (PED), when the device is initialized (or at a subsequent re-initialization following repair), the KRD vendor will inject data such that the KRD can be reliably distinguished from a counterfeit device. This data typically consists of a certificate, which corresponds to the private key of the KRD. The KRD's key pair may be sent to the KRD encrypted with a transport key or the KRD may self-generate the key pair and export a certificate signing request (CSR) to the CA and the CA responds with the certificate.

The KRD vendor will also inject the device with a CA certificate such that the KRD can validate KDH certificates issued by the same CA.

Phase 2 – KDH initialization: Prior to implementing SKDAT, the same CA certificate that was provided to the KRD is provided to the KDH. Also, similar to what happened in Phase 1 for the KRD, a KDH-specific key pair is generated by or installed (encrypted) into the HSM; the public key of that key pair is signed by the CA and the signed public key is installed in the KDH (host system, not in the SCD).

Phase 3 – Key exchange: When the KRD interfaces with a KDH, the KRD and KDH will exchange certificates. Each will use the CA certificate to verify the authenticity of the received certificate.

4.9.1.3.2 Phase 1 requirements

In Phase 1, the KRD public key that has not yet been signed shall be protected against substitution or modification. Sufficient controls shall exist to ensure that it is infeasible for a counterfeit device to be initialized such that it cannot be reliably distinguished from a genuine device. The CA shall have mechanisms to ensure that the CSR is received from a valid device. One method of achieving this is to contain the operations where the unsigned public key is presented to the CA for signature. If this

method is used, the activity to sign the public key shall occur within a secure environment as defined in the [ISO 13491](#) series.

Where the root level CA is used to sign a subordinate CA that issues certificates to a KDH or KRD, the authenticity and integrity of the root CA certificate installed with the signed sub-CA public key shall be verified prior to installation in the secure environment. The integrity of the CA certificate shall be maintained prior to loading into the KRD. Once the CA certificate is loaded into the KRD, it shall be protected against unauthorized modification, substitution or replay.

The request to generate or sign the KRD certificate shall include sufficient data for the CA to validate the authenticity of the request. The request channel to the CA shall be protected against unauthorized substitution, modification and replay. All requests to the CA to generate or sign certificates shall be logged by the requesting entity, including any requests that fail. The CA shall log all requests to generate or sign KRD certificates, retaining all element in the CSR, and enough information about the event to facilitate a forensic investigation.

4.9.1.3.3 Phase 2 requirements

The CA shall validate the KDH CSR in accordance with the registration authority requirements as defined in [4.4](#). The KDH certificate request to the CA should be sent through an integrity protected channel.

The CA public key shall be distributed as per the appropriate subclause of [ISO 21188](#).

The integrity of the CA certificate shall be verified prior to use by the KDH HSM. Once the CA certificate is loaded into the KDH, it shall be protected against unauthorized modification, substitution or replay.

Signing of the KDH public key shall be performed in a secure environment as defined in the [ISO 13491](#) series.

Procedural controls shall ensure that only authorized signed KDH public key(s) are installed for use.

4.9.1.3.4 Phase 3 requirements

In Phase 3, prior to transport of a symmetric key to the KRD, the KDH shall verify that this is the intended KRD (i.e. the KDH has prior knowledge of the identity of the KRD and does not rely only on the validity of the KRD certificate).

The KRD shall validate the legitimacy of the KDH prior to accepting the symmetric key and installing it for future use.

See [Annex C](#) for an overview of asymmetric distribution techniques.

4.9.1.4 Cleartext key distribution

4.9.1.4.1 General

Cleartext symmetric keys are distributed as components or shares of the key, or by using an SCD to transport the entire cleartext key.

Regardless of the key distribution method used to transport cleartext components or shares, it shall not be possible for any person to ascertain any of the bits of the final key during the distribution process.

In some cases, it is expedient to have authorized individuals carry (i.e. convey) components or shares of the key from one place to another (e.g. from the point of storage to the point of loading). Requirements for this type of situation are covered in [4.9.1.3.1](#). Subclause [4.9.1.4.3](#) and [Annex B](#) describe the requirements that apply when the transportation of the component(s) or share(s) is not under the physical control of authorized persons at all times (e.g. when mailing).

4.9.1.4.2 Personal conveyance of cleartext components or shares

When conveying keys, each cleartext component or share not in an SCD shall be in the physical control of only the authorized Custodian and should be in the possession of the authorized Custodian only for the minimum practical time required to transport the component or share, enter it into an SCD (if applicable) and then either securely store or destroy it.

The authorized custodian physically carrying a component or share in printed form shall ensure that the component or share value is not visible to any unauthorized individual during conveyance.

4.9.1.4.3 Transporting cleartext components or shares in an SCD

Cleartext key components or shares may be transported in one or more SCDs. This subclause covers the transport of a single component or share of one or more keys using an SCD (e.g. a single SCD containing "Custodian 1" components for one or more keys). Transporting multiple components or shares of a given key within a single SCD shall be considered equal to transporting a cleartext key within the SCD and is covered in [4.2](#).

The management of a component or share (including the SCD in which the component or share resides) shall ensure dual control and split knowledge of the resulting key.

For cleartext components or shares transported within an SCD, that SCD shall meet the following requirements in addition to those in [4.2](#):

- a) have logical controls to limit access to authorized individual(s);
- b) have logical controls that include protections against attacks (e.g. repeated failed login attempts);
- c) support authentication controls that render the device or user account unusable and/or remove the keys when the number of invalid authentication attempts exceeds a predefined limit.

The process to transport the SCD shall conform to [Annex B](#).

4.9.1.4.4 Transporting cleartext components or shares in printed form or on transport media

Transport media refers to any device that is used to contain a component or share that does not meet the criteria of an SCD (as defined in [4.2](#)), such as an HMD.

The sender and receiver have equal responsibility for the secrecy of the cleartext components or shares that are transported in printed form or on transport media. Both parties shall have procedures to ensure the secure handling of the components or shares throughout the process in accordance with the principles of dual control and split knowledge as described in [4.1.2](#).

The process to transport key components or shares that are in printed form or contained in transport media shall conform to [Annex B](#).

4.9.2 SKDAT asymmetric key distribution

4.9.2.1 Asymmetric key pair transfer

The asymmetric key pair transfer is the process by which the key pair and the certificate of the public key are conveyed to the owner of the key pair. This process occurs when the owner does not have the capacity to generate their key pair.

The owner shall be authenticated prior to being given their key pair.

The techniques used for public key distribution are described in [4.9.2.2](#) and plaintext private key distribution is covered in [4.9.2.3](#).

When the key pair is not generated by the key owner, prior to the distribution of the public key the correct transfer of the key pair should be verified.

4.9.2.2 Public key distribution

The public key of an asymmetric key pair needs to be distributed to, and stored by, one or more users for subsequent use as an encryption key and/or signature verification key, or for use in a key agreement mechanism. Public key distribution is the process by which a public key is conveyed to the party intended to use it. Although this key need not be protected from disclosure, during distribution to authorized recipients or during storage in a key database, it is necessary to ensure the authenticity of the public key.

Mechanisms for the distribution of asymmetric public keys are described in [ISO/IEC 11770-3](#).

Any distribution method (manual or automated) shall ensure the integrity and authenticity of the public key. The substitution of a public key during distribution shall be prevented. This can be achieved by maintaining the public key in the forms described in [4.1.3.2.2](#).

An independent communication shall be used to verify that the identification of the key and its owner are correct and authorized. This might require confirmation obtained via a different channel from the one whereby the information was originally obtained.

One or more of the following techniques shall be used to ensure public key integrity:

- sign the public key and associated data using a digital signature system, thereby creating a public key certificate in accordance with [4.3](#) and [4.4](#);
- create a MAC for the public key and associated data, using an algorithm defined by [ISO 16609](#) and a key used only for this purpose;
- store the public key in an SCD (see the [ISO 13491](#) series);
- distribute the public key over a channel that protects both the integrity and authenticity of the public key (e.g. VPN);
- distribute the public key over an unprotected channel and distribute a KCV of the public key and associated data over a channel that protects both the integrity and authenticity of the KCV; asymmetric KCVs are described in [4.8.3](#);
- use authenticated encryption;
- transport within a CSR.

4.9.2.3 Private key distribution

If it is necessary to output the private key from the generating SCD (e.g. for transfer to another SCD where it is to be used or for backup purposes), it shall be protected from compromise by at least one of the following techniques:

- encipherment with another cryptographic key;
- if non-encrypted and outside an SCD, as key shares using an acceptable key segmentation algorithm (see [ISO/IEC 19592-2](#)).

One or more of the following techniques shall be used to ensure private key integrity:

- Create a MAC for the private key and associated data, using an algorithm defined by [ISO 16609](#) and a key used only for this purpose.
- Use authenticated encryption.
- Use key blocks as defined in [ISO 20038](#) for AES or ANSI X9.143 for TDEA.
- Store as integrity-protected key shares.

- Verify that the private key and the authenticated public key form a valid key pair by first applying a transformation on arbitrary data using one of the keys, then applying the complementary transformation using the other key and then confirming the result against the expected result. This operation shall be wholly conducted within an SCD and all intermediate and final results of the transformation destroyed.

Key shares that will form a private key shall be distributed in conformity with the requirements for symmetric key shares. See [4.9](#).

The process of transferring encrypted private keys shall protect against key substitution and modification.

The key's identifier and related data should be transferred together with the private key.

Where the SCD that will use the private key is not the SCD that creates it, the private key shall not be recorded or retained by the source SCD.

4.10 Key loading

4.10.1 General

The loading of keys into an SCD shall be performed using one of following techniques:

- manual entry (i.e. typing cleartext components);
- electronic direct loading (e.g. using a KLD or an HMD);
- distribution over a network (i.e. SKDAT or master key or session key).

The permissible techniques to load keys into an SCD as a function of the different key forms are indicated in [Table 2](#).

Table 2 — Permissible key distribution and loading techniques

Key forms	Manual	Techniques	
		Electronic	
		Direct	Network
Plaintext keys	Not permitted	Not permitted	Not permitted
Plaintext key components or shares	Permitted	Permitted	Not permitted
Encrypted keys	Permitted	Permitted	Permitted
Encrypted key components or shares	Permitted	Permitted	Permitted

Regardless of the key loading method used, it shall be infeasible for any person to ascertain all or part of any final key during the key loading process.

Keys shall be loaded into an SCD only when it can be ensured that the device has not been subject to prior tampering which might lead to the disclosure of keys or sensitive data.

Except for sending encrypted values to an SCD that is housed in at least a controlled environment as defined in the [ISO 13491](#) series, the key loading process shall be performed according to the principles of dual control and split knowledge.

A key shall not be reloaded to a transaction-originating SCD, except by chance.

To facilitate the detection of errors in key loading, the check value for key components and keys should be verified.