BS ISO/IEC 11770-3:2021

BSI Standards Publication

# Information security — Key management

Part 3: Mechanisms using asymmetric techniques

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 11770-3:2021

# National foreword

This British Standard is the UK implementation of ISO/IEC 11770-3:2021. It supersedes BS ISO/IEC 11770-3:2015+A1:2017, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/33, Information security, cybersecurity and privacy protection.

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 2021
Published by BSI Standards Limited 2021

ISBN 978 0 539 17234 8

ICS 35.030

**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 30 November 2021.

**Amendments/corrigenda issued since publication**

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

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 11770-3:2021

INTERNATIONAL STANDARD

# ISO/IEC 11770-3

Fourth edition
2021-10

# Information security — Key management —

## Part 3: Mechanisms using asymmetric techniques

*Sécurité de l'information — Gestion de clés —*

*Partie 3: Mécanismes utilisant des techniques asymétriques*

Reference number
ISO/IEC 11770-3:2021(E)

Documents
Document Sharing Hub
© ISO/IEC 2021

Full document details: BS ISO/IEC 11770-3:2021

# Contents

Page


**Foreword** ..... 5

**Introduction** ..... 6

**1 Scope** ..... 1

**2 Normative references** ..... 2

**3 Terms and definitions** ..... 2

**4 Symbols and abbreviations** ..... 8

**5 Requirements** ..... 10

**6 Key derivation functions** ..... 11

**7 Cofactor multiplication** ..... 11

**8 Key commitment** ..... 12

**9 Key confirmation** ..... 12

**10 Framework for key management** ..... 13
**10.1 General** ..... 13
**10.2 Key agreement between two parties** ..... 14
**10.3 Key agreement between three parties** ..... 14
**10.4 Secret key transport** ..... 15
**10.5 Public key transport** ..... 15

**11 Key agreement** ..... 15
**11.1 Key agreement mechanism 1** ..... 15
**11.2 Key agreement mechanism 2** ..... 17
**11.3 Key agreement mechanism 3** ..... 17
**11.4 Key agreement mechanism 4** ..... 19
**11.5 Key agreement mechanism 5** ..... 20
**11.6 Key agreement mechanism 6** ..... 21
**11.7 Key agreement mechanism 7** ..... 23
**11.8 Key agreement mechanism 8** ..... 24
**11.9 Key agreement mechanism 9** ..... 25
**11.10 Key agreement mechanism 10** ..... 26
**11.11 Key agreement mechanism 11** ..... 27
**11.12 Key agreement mechanism 12** ..... 28
**11.13 Key agreement mechanism 13** ..... 29
**11.14 Key agreement mechanism 14** ..... 30
**11.15 Key agreement mechanism 15** ..... 31

**12 Secret key transport** ..... 32
**12.1 Secret key transport mechanism 1** ..... 32
**12.2 Secret key transport mechanism 2** ..... 34
**12.3 Secret key transport mechanism 3** ..... 35
**12.4 Secret key transport mechanism 4** ..... 37
**12.5 Secret key transport mechanism 5** ..... 38
**12.6 Secret key transport mechanism 6** ..... 41

**13 Public key transport** ..... 42
**13.1 Public key transport mechanism 1** ..... 42
**13.2 Public key transport mechanism 2** ..... 43
**13.3 Public key transport mechanism 3** ..... 44


Documents
Document Sharing Hub

© ISO/IEC 2021 – All rights reserved


Full document details: BS ISO/IEC 11770-3:2021

**Annex A** (normative) **Object identifiers** ..... **46**
**Annex B** (informative) **Properties of key establishment mechanisms** ..... **55**
**Annex C** (informative) **Examples of key derivation functions** ..... **58**
**Annex D** (informative) **Examples of key establishment mechanisms** ..... **66**
**Annex E** (informative) **Examples of elliptic curve based key establishment mechanisms** ..... **70**
**Annex F** (informative) **Example of bilinear pairing based key establishment mechanisms** ..... **80**
**Annex G** (informative) **Secret key transport** ..... **84**
**Bibliography** ..... **88**


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

# Foreword

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

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

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

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

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

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *Information security, cybersecurity and privacy protection*.

This fourth edition cancels and replaces the third edition (ISO/IEC 11770-3:2015), which has been technically revised. It also incorporates Technical Corrigenda ISO/IEC 11770-3:2015/Cor1:2016 and ISO/IEC 11770-3:2015/Amd.1:2017.

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

— the blinded Diffie-Hellman key agreements are added as key agreement mechanism 13 and 14 and examples of the mechanisms are included in Annex E;

— key agreement mechanism 15 is added and the SM9 key agreement protocol as an example of the mechanism is included in Annex F.

A list of all parts in the ISO/IEC 11770 series can be found on the ISO and IEC websites.

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

© ISO/IEC 2021 – All rights reserved


Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

# Introduction

This document describes schemes that can be used for key agreement and schemes that can be used for key transport.

Public key cryptosystems were first proposed in the seminal paper by Diffie and Hellman in 1976. The security of many such cryptosystems is based on the presumed intractability of solving the discrete logarithm problem over certain finite fields. Other public key cryptosystems such as RSA are based on the difficulty of the integer factorization problem.

A third class of public key cryptosystems is based on elliptic curves. The security of such a public key system depends on the difficulty of determining discrete logarithms in the group of points of an elliptic curve. When based on a carefully chosen elliptic curve, this problem is, with current knowledge, much harder than the factorization of integers or the computation of discrete logarithms in a finite field of comparable size. All known general purpose algorithms for determining elliptic curve discrete logarithms take exponential time. Thus, it is possible for elliptic curve based public key systems to use much shorter parameters than the RSA system or the classical discrete logarithm based systems that make use of the multiplicative group of some finite field. This yields significantly shorter digital signatures, as well as system parameters, and allows for computations using smaller integers.

This document includes mechanisms based on the following:

— finite fields;

— elliptic curves;

— bilinear pairings.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents and http://patents.iec.ch.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

# Information security — Key management —

## Part 3:
## Mechanisms using asymmetric techniques

## 1 Scope

This document defines key management mechanisms based on asymmetric cryptographic techniques. It specifically addresses the use of asymmetric techniques to achieve the following goals.

a) Establish a shared secret key for use in a symmetric cryptographic technique between two entities *A* and *B* by key agreement. In a secret key agreement mechanism, the secret key is computed as the result of a data exchange between the two entities *A* and *B*. Neither of them is able to predetermine the value of the shared secret key.

b) Establish a shared secret key for use in a symmetric cryptographic technique between two entities *A* and *B* via key transport. In a secret key transport mechanism, the secret key is chosen by one entity *A* and is transferred to another entity *B*, suitably protected by asymmetric techniques.

c) Make an entity's public key available to other entities via key transport. In a public key transport mechanism, the public key of entity *A* is transferred to other entities in an authenticated way, but not requiring secrecy.

Some of the mechanisms of this document are based on the corresponding authentication mechanisms in ISO/IEC 9798-3.

This document does not cover certain aspects of key management, such as:

— key lifecycle management;

— mechanisms to generate or validate asymmetric key pairs; and

— mechanisms to store, archive, delete, destroy, etc., keys.

While this document does not explicitly cover the distribution of an entity's private key (of an asymmetric key pair) from a trusted third party to a requesting entity, the key transport mechanisms described can be used to achieve this. A private key can in all cases be distributed with these mechanisms where an existing, non-compromised key already exists. However, in practice the distribution of private keys is usually a manual process that relies on technological means such as smart cards, etc.

This document does not specify the transformations used in the key management mechanisms.

NOTE To provide origin authentication for key management messages, it is possible to make provisions for authenticity within the key establishment protocol or to use a public key signature system to sign the key exchange messages.

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

## 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 10118-1, *Information technology — Security techniques — Hash-functions — Part 1: General*

ISO/IEC 11770-1, *Information technology — Security techniques — Key management — Part 1: Framework*

ISO/IEC 11770-6, *Information technology — Security techniques — Key management — Part 6: Key derivation*

ISO/IEC 15946-1, *Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Part 1: General*

ISO/IEC 18031, *Information technology — Security techniques — Random bit generation*

ISO/IEC 19772, *Information security — Authenticated encryption*

## 3 Terms and definitions

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

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

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

**3.1**
**asymmetric cryptographic technique**
cryptographic technique that uses two related transformations, a public transformation [defined by the *public key* (3.33)] and a private transformation [defined by the *private key* (3.32)], and has the property that given the public transformation, then it is computationally infeasible to derive the private transformation

Note 1 to entry: A system based on asymmetric cryptographic techniques can either be an encryption system, a signature system, a combined encryption and signature system, or a key agreement scheme. With asymmetric cryptographic techniques there are four elementary transformations: *signature* and *verification* for signature systems, *encryption* and *decryption* for encryption systems. The signature and the decryption transformations are kept private by the owning entity, whereas the corresponding verification and encryption transformations are published. There exist asymmetric cryptosystems (e.g. RSA) where the four elementary functions can be achieved by only two transformations: one private transformation suffices for both signing and decrypting messages, and one public transformation suffices for both verifying and encrypting messages. However, since this does not conform to the principle of key separation, throughout this document the four elementary transformations and the corresponding keys are kept separate.

**3.2**
**asymmetric encryption system**
system based on *asymmetric cryptographic techniques* (3.1) whose public transformation is used for *encryption* (3.9) and whose private transformation is used for *decryption* (3.6)


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

**3.3**
**asymmetric key pair**
pair of related *keys* (3.17) where the *private key* (3.32) defines the private transformation and the *public key* (3.33) defines the public transformation

**3.4**
**certification authority**
**CA**
centre trusted to create and assign *public key* (3.33) certificates

**3.5**
**collision-resistant hash-function**
*hash-function* (3.15) satisfying the following property: it is computationally infeasible to find any two distinct inputs which map to the same output

Note 1 to entry: Computational feasibility depends on the specific security requirements and environment.

[SOURCE: ISO/IEC 10118-1:2016, 3.1]

**3.6**
**decryption**
reversal of a corresponding *encryption* (3.9)

[SOURCE: ISO/IEC 11770-1:2010, 2.6]

**3.7**
**digital signature**
data unit appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to verify the origin and integrity of the data unit and protect the sender and the recipient of the data unit against forgery by third parties, and the sender against forgery by the recipient

**3.8**
**distinguishing identifier**
information which unambiguously distinguishes an entity

[SOURCE: ISO/IEC 11770-1:2010, 2.9]

**3.9**
**encryption**
(reversible) transformation of data by a cryptographic algorithm to produce ciphertext, i.e. to hide the information content of the data

[SOURCE: ISO/IEC 11770-1:2010, 2.10]

**3.10**
**entity authentication**
corroboration that an entity is the one claimed

[SOURCE: ISO/IEC 9798-1:2010, 3.14]

**3.11**
**entity authentication of entity *A* to entity *B***
assurance of the identity of entity *A* for entity *B*

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

**3.12**
**explicit key authentication from entity *A* to entity *B***
assurance for entity *B* that entity *A* is the only other entity that is in possession of the correct *key* (3.17)

Note 1 to entry: Implicit key authentication from entity *A* to entity *B* and key confirmation from entity *A* to entity *B* together imply explicit key authentication from entity *A* to entity *B*.

**3.13**
**forward secrecy with respect to both entity *A* and entity *B* individually**
property that knowledge of entity *A*'s long-term *private key* (3.32) or knowledge of entity *B*'s long-term *private key* (3.32) subsequent to a *key agreement* (3.18) operation does not enable an opponent to recompute previously derived *keys* (3.17)

Note 1 to entry: This differs from mutual forward secrecy in which knowledge of both entity *A*'s and entity *B*'s long-term private keys do not enable recomputation of previously derived keys.

**3.14**
**forward secrecy with respect to entity *A***
property that knowledge of entity *A*'s long-term *private key* (3.32) subsequent to a *key agreement* (3.18) operation does not enable an opponent to recompute previously derived *keys* (3.17)

**3.15**
**hash-function**
function which maps strings of bits of variable (but usually upper bounded) length to fixed-length strings of bits, satisfying the following two properties:

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

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

Note 1 to entry: Computational feasibility depends on the specific security requirements and environment.

Note 2 to entry: For the purposes of this document all hash-functions are assumed to be collision-resistant hash-functions.

[SOURCE: ISO/IEC 10118-1:2016, 3.4]

**3.16**
**implicit key authentication from entity *A* to entity *B***
assurance for entity *B* that entity *A* is the only other entity that can possibly be in possession of the correct *key* (3.17)

**3.17**
**key**
sequence of symbols that controls the operation of a cryptographic transformation (e.g. *encryption* (3.9), *decryption* (3.6), cryptographic check function computation, signature calculation, or signature verification)

[SOURCE: ISO/IEC 11770-1:2010, 2.12]


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

**3.18**
**key agreement**
process of establishing a shared *secret key* (3.38) between entities in such a way that neither of them can predetermine the value of that *key* (3.17)
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.

**3.19**
**key commitment**
process of committing to use specific *keys* (3.17) in the operation of a *key agreement* (3.18) scheme before revealing the specified *keys* (3.17)

**3.20**
**key confirmation from entity *A* to entity *B***
assurance for entity *B* that entity *A* is in possession of the correct *key* (3.17)

**3.21**
**key control**
ability to choose the *key* (3.17) or the parameters used in the *key* (3.17) computation

**3.22**
**key derivation function**
function that outputs one or more shared secrets, for use as *keys* (3.17), given shared secrets and other mutually known parameters as input

**3.23**
**key establishment**
process of making available a shared *secret key* (3.38) to one or more entities, where the process includes *key agreement* (3.18) and *key transport* (3.25)

**3.24**
**key token**
*key* (3.17) management message sent from one entity to another entity during the execution of a *key* (3.17) management mechanism

**3.25**
**key transport**
process of transferring a *key* (3.17) from one entity to another entity, suitably protected

**3.26**
**message authentication code**
**MAC**
string of bits which is the output of a *MAC algorithm* (3.27)

Note 1 to entry: A MAC is sometimes called a cryptographic check value (see for example ISO 7498-2).

[SOURCE: ISO/IEC 9797-1:2011, 3.9]

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

**3.27**
**message authentication code algorithm**
**MAC algorithm**
algorithm for computing a function which maps strings of bits and a *secret key* (3.38) to fixed-length strings of bits, satisfying the following two properties:
— for any *key* (3.17) and any input string, the function can be computed efficiently;
— for any fixed *key* (3.17), and given no prior knowledge of the *key* (3.17), it is computationally infeasible to compute the function value on any new input string, even given knowledge of a set of input strings and corresponding function values, where the value of the *i*th input string can have been chosen after observing the value of the first $i - 1$ function values (for integers $i > 1$)

Note 1 to entry: A MAC algorithm is sometimes called a cryptographic check function (see for example ISO 7498-2).

Note 2 to entry: Computational feasibility depends on the user's specific security requirements and environment.

[SOURCE: ISO/IEC 9797-1:2011, 3.10]

**3.28**
**mutual entity authentication**
*entity authentication* (3.10) which provides both entities with assurance of each other's identity

**3.29**
**mutual forward secrecy**
property that knowledge of both entity *A*'s and entity *B*'s long-term *private keys* (3.32) subsequent to a *key agreement* (3.18) operation does not enable an opponent to recompute previously derived *keys* (3.17)

**3.30**
**one-way function**
function with the property that it is easy to compute the output for a given input but it is computationally infeasible to find an input which maps to a given output

**3.31**
**prefix free representation**
representation of a data element for which concatenation with any other data does not produce a valid representation

**3.32**
**private key**
*key* (3.17) of an entity's *asymmetric key pair* (3.3) that is kept private

Note 1 to entry: The security of an asymmetric system depends on the privacy of this key.

[SOURCE: ISO/IEC 11770-1:2010, 2.35]

**3.33**
**public key**
*key* (3.17) of an entity's *asymmetric key pair* (3.3) which can usually be made public without compromising security

Note 1 to entry: In the case of an asymmetric signature system, the public key defines the verification transformation. In the case of an asymmetric encryption system, the public key defines the encryption transformation, conditional on the inclusion of randomisation elements. A key that is "publicly known" is not necessarily globally available. The key can only be available to all members of a pre-specified group.

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


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

**3.34**
**public key certificate**
*public key information* (3.35) of an entity signed by the *certification authority* (3.4) and thereby rendered unforgeable

**3.35**
**public key information**
information containing at least the entity's *distinguishing identifier* (3.8) and *public key* (3.33), but can include other static information regarding the *certification authority* (3.4), the entity, restrictions on *key* (3.17) usage, the validity period, or the involved algorithms

**3.36**
**resilience to key compromise impersonation attack on *A***
resilience to attacks in which an adversary exploits knowledge of the long-term *private key* (3.32) of *A* to impersonate any entity in subsequent communication with *A*

**3.37**
**resilience to unknown key share attack for *A* and *B***
resilience to attacks in which only *A* and *B* know the session *key* (3.17) *K*; however, *A* and *B* disagree on who they share *K* with

Note 1 to entry: Resilience to unknown key share attack can be achieved by choosing a key derivation function that includes the identifiers of the involved entities.

**3.38**
**secret key**
*key* (3.17) used with symmetric cryptographic techniques by a specified set of entities

**3.39**
**sequence number**
*time variant parameter* (3.44) whose value is taken from a specified sequence which is non-repeating within a certain time period

[SOURCE: ISO/IEC 11770-1:2010, 2.44]

**3.40**
**signature system**
system based on *asymmetric cryptographic techniques* (3.1) whose private transformation is used for signing and whose public transformation is used for verification

**3.41**
**third party forward secrecy**
property that knowledge of a third party's *private key* (3.32) subsequent to a *key agreement* (3.18) operation does not enable an opponent to recompute previously derived *keys* (3.17)

Note 1 to entry: Instead of third party forward secrecy, master key forward secrecy is also used in Reference [19].

**3.42**
**time stamp**
data item which denotes a point in time with respect to a common time reference

**3.43**
**time-stamping authority**
*trusted third party* (3.45) trusted to provide a time-stamping service
[SOURCE: ISO/IEC 13888-1:2020, 3.54]

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

**3.44**
**time variant parameter**
data item used to verify that a message is not a replay, such as a random number, a *time stamp* (3.42) or a *sequence number* (3.39)

Note 1 to entry: If a random number is used. then this is as a challenge in a challenge-response protocol. See also ISO/IEC 9798-1:2010, Annex B.

[SOURCE: ISO/IEC 9798-1:2010, 3.36]

**3.45**
**trusted third party**
security authority or its agent, trusted by other entities with respect to security related activities

[SOURCE: ISO/IEC 9798-1:2010, 3.38]

## 4 Symbols and abbreviations

The following symbols and abbreviations are used in this document.

| | |
|---|---|
| $A$, $B$, $C$ | distinguishing identifiers of entities |
| $BE$ | encrypted data block |
| $BS$ | signed data block |
| BS2IP | bit string to integer conversion primitive |
| CA | certification authority |
| $Cert_A$ | entity $A$'s public key certificate |
| $D_A$ | entity $A$'s private decryption transformation function |
| $d_A$ | entity $A$'s private decryption key |
| $E$ | elliptic curve, either given by an equation of the form $Y^2 = X^3 + aX + b$ over the field GF($p^m$) for $p$>3 and a positive integer $m$, by an equation of the form $Y^2 + XY = X^3 + aX^2 + b$ over the field GF($2^m$), or by an equation of the form $Y^2 = X^3 + aX^2 + b$ over the field GF($3^m$), together with an extra point $O_E$ referred to as the point at infinity, which is denoted by $E$/GF($p^m$), $E$/GF($2^m$), or $E$/GF($3^m$), respectively |
| $E_A$ | entity $A$'s public encryption transformation function |
| $e_A$ | entity $A$'s public encryption key |
| F | key agreement function |
| F($h$,$g$) | key agreement function using as input a factor $h$ and a common element $g$ |
| FP | key agreement function based on pairing |
| $G$ | point on $E$ with order $n$ |
| $g$ | common element shared publicly by all the entities that use the key agreement function F |
| gcd($a$,$b$) | greatest common divisor of two integers $a$ and $b$ |
| GF($p^m$), GF($2^m$), GF($3^m$) | finite field with $p^m$ , $2^m$, $3^m$ elements for a prime $p$>3 and a positive integer $m$ |
| $H_A$ | private key agreement key in a pairing-friendly elliptic curve of entity $A$ |


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

| | |
|---|---|
| $h_A$ | entity $A$'s private key agreement key |
| hash | hash-function |
| IHF2($A$, $n$) | BS2IP(kdf($A$, $8\lceil(5\ b_n)/32\rceil$)) mod ($n$-1) +1 where $b_n$ is the bit-length of $n$ |
| $j$ | cofactor used in performing cofactor multiplication |
| $K$ | secret key for a symmetric cryptosystem |
| $K_{AB}$ | secret key shared between entities $A$ and $B$ |

NOTE 1 In practical implementations, the shared secret key is subject to further processing before it can be used for a symmetric cryptosystem.

| | |
|---|---|
| kdf | key derivation function |
| $KT$ | key token |
| $KT_A$ | entity $A$'s key token |
| $KT_{Ai}$ | key token sent by entity $A$ after step ($Ai$) |
| $l$ | supplementary value used in performing cofactor multiplication |
| lcm | least common multiple |
| $M$ | data message |
| MAC | Message Authentication Code |
| $\mathrm{MAC}_K(Z)$ | output of a MAC algorithm when using as input the secret key $K$ and an arbitrary data string $Z$ |
| MQV | Menezes-Qu-Vanstone |
| $n$ | prime divisor of the order (or cardinality) of an elliptic curve $E$ over a finite field |
| $O_E$ | elliptic curve point at infinity |
| $P$ | point on an elliptic curve $E$ |
| $P_A$ | public key-agreement key in an elliptic curve of entity $A$ |
| $p_A$ | entity $A$'s public key-agreement key |
| pairing | pairing defined over an elliptic curve and used in FP |
| *parameters* | parameters used in the key derivation function |
| $PKI_A$ | entity $A$'s public key information |
| $q$ | prime power $p^m$ for some prime $p \neq 3$ and some integer $m \geq 1$ |
| $r$ | random number generated in the course of a mechanism |
| $r_A$ | random number issued by entity $A$ in a key agreement mechanism |
| $S_1, S_2, S_3$ | sets of elements |
| $\mathrm{S}_A$ | entity $A$'s private signature transformation function |
| $s_A$ | entity $A$'s private signature key |
| $T$ | trusted third party |
| *Texti* | $i$th optional text, data or other information that may be included in a data block, if desired |

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

| | |
|---|---|
| $TVP$ | time-variant parameter such as a random number, a time stamp, or a sequence number |
| $V_A$ | entity $A$'s public verification transformation function |
| $v_A$ | entity $A$'s public verification key |
| w | one-way function |
| $X(P)$ | x-coordinate of a point P |
| $\sqrt{q}$ | square root of a positive number $q$ |
| $\#E$ | order (or cardinality) of an elliptic curve $E$ |
| $\|\|$ | concatenation of two data elements |
| $\lceil x \rceil$ | smallest integer greater than or equal to the real number $x$ |
| $\Sigma$ | digital signature |
| $\pi(P)$ | $(X(P) \bmod 2^{\lceil \rho/2 \rceil}) + 2^{\lceil \rho/2 \rceil}$ where $\rho = \lceil \log_2 n \rceil$ and $X(P)$ is the $x$-coordinate of the point $P$ |

NOTE 2 No assumption is made on the nature of the signature transformation. In the case of a signature system with message recovery, $S_A(M)$ denotes the signature $\Sigma$ itself. In the case of a signature system with appendix, $S_A(M)$ denotes the message $M$ together with the signature $\Sigma$.

NOTE 3 The keys of an asymmetric cryptosystem are denoted by lower case letters (indicating its function) indexed with the identifier of its owner, e.g. the public verification key of entity $A$ is denoted by $v_A$. The corresponding transformations are denoted by upper case letters indexed with the identifier of their owner, e.g. the public verification transformation of entity $A$ is denoted by $V_A$.

## 5 Requirements

It is assumed that the entities involved in a mechanism are aware of each other's claimed identities. This can be achieved by the inclusion of identifiers in information exchanged between the two entities, or it can be apparent from the context of use of the mechanism. Verifying the identity means checking that a received identifier field agrees with some known (trusted) or expected value.

If a public key is registered with an entity, then that entity shall make sure that the entity who registers the key is in possession of the corresponding private key (see ISO/IEC 11770-1 for further guidance on key registration).

Annex A lists the object identifiers which shall be used to identify the key management mechanisms specified in this document.

Annex B summarizes the major properties of the key establishment/transport mechanisms specified in this document.

Annex C provides examples of key derivation functions.

Annex D provides examples of key establishment mechanisms.

Annex E provides examples of elliptic curve based key establishment mechanisms.

Annex F provides example of bilinear pairing based key establishment mechanisms.

Annex G describes secret key transport.


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

# 6 Key derivation functions

The use of a shared secret as derived in Clause 10 as a key for a symmetric cryptosystem without further processing is not recommended. It is often the case that the form of a shared secret established as a result of using a mechanism specified in this document does not conform to the form needed for a specific cryptographic algorithm, so some processing is needed. Moreover, the shared secret (often) has arithmetic properties and relationships that can result in a shared symmetric key not being chosen from the full key space. It is therefore advisable to pass the shared secret through a key derivation function, e.g. involving the use of a hash function. The use of an inadequate key derivation function can compromise the security of the key agreement scheme with which it is used. It is recommended to use a one-way function as a key derivation function.

A key derivation function produces keys that are computationally indistinguishable from randomly generated keys. The key derivation function takes as input a shared secret and a set of key derivation parameters and produces an output of the desired length.

In order for the two parties in a key establishment mechanism to agree on a common secret key, the key derivation function shall be agreed (see ISO/IEC 11770-6 for further guidance on key derivation functions).

Annex C provides examples of key derivation functions.

# 7 Cofactor multiplication

This clause applies only to mechanisms using elliptic curve cryptography. The key agreement mechanisms in Clause 11 and the key transport mechanisms in Clauses 12 and 13 require that the user's private key or key token be combined with another entity's public key or key token. If the other entity's public key or key token is not valid (i.e. it is not a point on the elliptic curve, or is not in the subgroup of order $n$), then performing this operation can result in some bits of the private key being leaked to an attacker. One example of such an attack is known as the "small subgroup attack".

NOTE 1 The small subgroup attack is described in Reference [37].

In order to prevent the "small subgroup attack" and similar attacks, one option is to validate public keys and key tokens received from the other party using public key validation, as specified in ISO/IEC 11770-1.

As an alternative to public key validation, a technique called cofactor multiplication as specified in Clause 11 can be used. The values $j$ and $l$, defined below, are used in cofactor multiplication.

If cofactor multiplication is used, there are two options:

— if compatibility with entities not using cofactor multiplication is not required, then let $j = \#E / n$ and $l = 1$. If this option is chosen, both parties involved shall agree to use this option, otherwise the mechanism will not work;

— if compatibility with entities not using cofactor multiplication is required, then let $j = \#E / n$ and $l = j^{-1} \bmod n$.

NOTE 2 The value $j^{-1} \bmod n$ always exists since $n$ is required to be greater than $4\sqrt{q}$ and therefore $\gcd(n, j) = 1$.

If cofactor multiplication is not required, then let $j = l = 1$.

Regardless of whether or not cofactor multiplication is used, if the shared key (or a component of the shared key) evaluates to the point at infinity ($O_E$), then the user shall assume that the key agreement procedure has failed.

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

It is particularly appropriate to perform public key validation or cofactor multiplication in the following cases:

— if the entity's public key is not authenticated;

— if the key token is not authenticated;

— if the user's public key is intended for a long-term use.

If the other entity's public key is authenticated and the cofactor is small, then the amount of information that can be leaked is limited. Thus, it is not always necessary to perform these tests.

## 8 Key commitment

Clause 11 describes key agreement mechanisms in which the established key is the result of applying a one-way function to the private key-agreement keys. However, it is possible that one entity knows the other entity's public key or key token prior to choosing their private key. As a result, such an entity can control the value of $s$ bits in the established key, at the cost of generating $2^s$ candidate values for their private key-agreement key in the time interval between discovering the other entity's public key or key token and choosing their own private key[31].

One way to address this concern (if it is a concern) at the cost of one additional message/pass in the protocol is through the use of key commitment. Key commitment can be performed by having the first entity hash the public key or key token and send the hash-code to the second entity. The second entity then replies with its public key or key token, and the first entity replies with its public key or key token. The second entity can now hash it and verify that the result is equal to the hash-code sent earlier.

## 9 Key confirmation

Explicit key confirmation is the process of adding additional messages to a key establishment protocol providing implicit key authentication, so that explicit key authentication and entity authentication are provided. Explicit key confirmation can be added to any method that does not possess it inherently. Key confirmation is typically provided by exchanging a value that can (with very high probablity) only be calculated correctly if the key establishment calculations were successful. Key confirmation from entity $A$ to entity $B$ is provided by entity $A$ calculating a value and sending it to entity $B$ for confirmation of entity $A$'s correct calculation. If mutual key confirmation is desired, then each entity sends a different value to the other.

Key confirmation is often provided by subsequent use of an established key, and if something is wrong then it is immediately detected. This is called implicit key confirmation. Explicit key confirmation in this case can be unnecessary. If one entity is not online (for example, in one-pass protocols used in store and forward (email) scenarios), then it is simply not possible for the other entity to obtain key confirmation. However, sometimes a key is established yet used only later (if at all), or the entity performing the key establishment process can simply not know if the resulting key will be used immediately or not. In these cases, it is often desirable to use a method of explicit key confirmation, as it can otherwise be too late to correct an error once detected. Explicit key confirmation can also be seen as a way of "firming up" security properties during the key establishment process and can be warranted if a conservative protocol design is deemed appropriate.

An example method of providing key confirmation using a MAC is as follows:

Entities $A$ and $B$ first perform one of the key establishment procedures specified in Clauses 11 and 12 of this document. As a result, they expect to share a secret MAC key $K_{AB}$. They then perform the following procedure.


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

— Entity $B$ forms the message $M$, an octet string consisting of the message identifier octet 0x02, entity $B$'s identifier, entity $A$'s identifier, the octet string $KT_B$ corresponding to entity $B$'s key token (omitted if not present), the octet string $KT_A$ corresponding to entity $A$'s key token (omitted if not present), the octet string $p_B$ corresponding to entity $B$'s public key-establishment key (omitted if not present), the octet string $p_A$ corresponding to entity $A$'s public key-establishment key (omitted if not present) and, if present, optional additional $Text1$, i.e.:

$M = 02||B||A||KT_B||KT_A||p_B||p_A||Text1$, where 0x02 is the message number.

— Entity $B$ calculates $K_B = \text{kdf}(K_{AB})$, and then calculates $\text{MAC}_{K_B}(M)$ for the message $M$ under the (supposedly) shared secret key $K_B$ for an appropriate MAC scheme.

— Entity $B$ sends the message $M$ and $\text{MAC}_{K_B}(M)$ to entity $A$.

— Entity $A$ calculates $K_A = \text{kdf}(K_{AB})$, computes $\text{MAC}_{K_A}(M)$ using the received message $M$, and verifies $\text{MAC}_{K_B}(M) = \text{MAC}_{K_A}(M)$.

— Assuming the MAC verifies, entity $A$ has received key confirmation from entity $B$ (that is, entity $A$ knows that $K_A$ equals $K_B$). If mutual key confirmation is desired, entity $A$ continues the protocol and forms the message $M'$ as the octet string consisting of the message identifier octet 0x03, entity $A$'s identifier, entity $B$'s identifier, the octet string $KT_A$ corresponding to entity $A$'s key token (omitted if not present), the octet string $KT_B$ corresponding to entity $B$'s key token (omitted if not present), the octet string $p_A$ corresponding to entity $A$'s public key-establishment key (omitted if not present), the octet string $p_B$ corresponding to entity $B$'s public key-establishment key (omitted if not present) and optional additional octet string $Text2$, i.e.:

$M' = 03||A||B||KT_A||KT_B||p_A||p_B||Text2$, where 0x03 is the message number.

— Entity $A$ calculates $\text{MAC}_{K_A}(M')$ under the (supposedly) shared secret $K_A$ using an appropriate MAC scheme.

— Entity $A$ sends $M'$ and $\text{MAC}_{K_A}(M')$ to entity $B$.

— Entity $B$ uses $K_B$ to verify $\text{MAC}_{K_A}(M')$ on the message $M'$. Assuming the MAC verifies, entity $B$ has received key confirmation from entity $A$ (that is, entity $B$ knows that $K_A$ equals $K_B$).

Other methods of key confirmation are possible. If the shared secret is to be used for data confidentiality (encryption), one entity can send the encryption of some specific plaintext known to the other entity, for example a block of all binary zeros or all binary ones. Care should be taken that any subsequent use of the key is very unlikely to encrypt the same plaintext as was used for key confirmation.

# 10 Framework for key management

## 10.1 General

This clause contains a high-level description of a framework for the key establishment mechanisms specified in this document. Four categories of mechanism are defined (key agreement between two parties, key agreement between three parties, secret key transport and public key transport), together with requirements for their use.

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

## 10.2 Key agreement between two parties

The provisions in this subclause apply to key agreement mechanisms 11.1 to 11.11 and 11.13 to 11.15, all of which specify mechanisms for key agreement between two parties. Key agreement between two parties is the process of establishing a shared secret key between two entities $A$ and $B$ in such a way that neither of them can predetermine the value of the shared secret key. Key agreement mechanisms can provide for implicit key authentication; in the context of key establishment, implicit key authentication means that after the execution of the mechanism only an identified entity can be in possession of the correct shared secret key.

Key agreement between two entities $A$ and $B$ takes place in a context shared by the two entities. The context consists of sets $S_1$ and $S_2$, and a key agreement function F. The function F shall satisfy the following requirements.

a) $\mathrm{F} : S_1 \times S_2 \rightarrow S_2$ maps elements $(h, g) \in S_1 \times S_2$ to $S_2$, and let $y = \mathrm{F}(h, g)$.

b) F satisfies the commutativity condition $\mathrm{F}(h_A, \mathrm{F}(h_B, g)) = \mathrm{F}(h_B, \mathrm{F}(h_A, g))$.

c) It is computationally intractable to find $\mathrm{F}(h_1, \mathrm{F}(h_2, g))$ from $\mathrm{F}(h_1, g)$, $\mathrm{F}(h_2, g)$ and $g$. This implies that $\mathrm{F}(\cdot, g)$ is a one-way function.

d) The entities $A$ and $B$ share a common element $g$ in $S_2$ which can be publicly known.

e) The entities acting in this setting can efficiently compute function values $\mathrm{F}(h, g)$ and can efficiently generate random elements in $S_1$. Depending on the particular key agreement mechanism, further conditions can be imposed.

NOTE 1 Examples for the function F are given in Annex D and Annex E. See also ISO/IEC 15946-1.

NOTE 2 As discussed in Clause 6, in practical implementations of the key agreement mechanisms the shared secret key is subject to further processing.

NOTE 3 It is in general necessary to check the received function values $\mathrm{F}(h, g)$ for weak values. If such values are encountered, the protocol aborts.

## 10.3 Key agreement between three parties

This clause applies to the key agreement mechanism 11.12 that describes key agreement between three parties. Key agreement between three parties is the process of establishing a shared secret key among three entities $A$, $B$, and $C$ in such a way that none of them can predetermine the value of the shared secret key. Key agreement among three entities $A$, $B$, and $C$ takes place in a context shared by the three entities. The context consists of sets $S_1$, $S_2$, and $S_3$, a function F, and a function FP. The functions F and FP shall satisfy the following requirements.

— $\mathrm{F} : S_1 \times S_2 \rightarrow S_2$ maps elements $(h, g) \in S_1 \times S_2$ to $S_2$, and let $y = \mathrm{F}(h, g)$.

— F satisfies the commutativity condition $\mathrm{F}(h_A, \mathrm{F}(h_B, g)) = \mathrm{F}(h_B, \mathrm{F}(h_A, g))$.

— It is computationally intractable to find $\mathrm{F}(h_1, \mathrm{F}(h_2, g))$ from $\mathrm{F}(h_1, g)$, $\mathrm{F}(h_2, g)$ and $g$. This implies that $\mathrm{F}(\cdot, g)$ is a one-way function.

— $\mathrm{FP} : S_1 \times S_2 \times S_2 \rightarrow S_3$ maps an element $(h_C, \mathrm{F}(h_A, g), \mathrm{F}(h_B, g)) \in S_1 \times S_2 \times S_2$ to an element of $S_3$, and let $z = \mathrm{FP}(h_C, \mathrm{F}(h_A, g), \mathrm{F}(h_B, g))$. ISO/IEC 15946-1 shall be referred for the relation between F and FP.

— FP satisfies the commutativity condition

  — $\mathrm{FP}(h_C, \mathrm{F}(h_A, g), \mathrm{F}(h_B, g)) = \mathrm{FP}(h_C, \mathrm{F}(h_B, g), \mathrm{F}(h_A, g)) = \mathrm{FP}(h_B, \mathrm{F}(h_A, g), \mathrm{F}(h_C, g))$


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

$$= \mathrm{FP}(h_A, \mathrm{F}(h_B, g), \mathrm{F}(h_C, g)) = \mathrm{FP}(h_A, \mathrm{F}(h_C, g), \mathrm{F}(h_B, g)) = \mathrm{FP}(h_B, \mathrm{F}(h_C, g), \mathrm{F}(h_A, g)).$$

— It is computationally intractable to find $\mathrm{FP}(h_C, \mathrm{F}(h_A, g), \mathrm{F}(h_B, g))$ from $\mathrm{F}(h_A, g)$, $\mathrm{F}(h_B, g)$, $\mathrm{F}(h_C, g)$, and $g$. This implies that $\mathrm{F}(\cdot, p_A, p_B)$ is a one-way function.

— The entities $A$, $B$, and $C$ share a common element $g$ in $S_2$ which can be publicly known.

— The entities acting on this setting can efficiently compute function values $\mathrm{F}(h, g)$ and $\mathrm{FP}(h_C, \mathrm{F}(h_B, g), \mathrm{F}(h_A, g))$, and can efficiently generate random elements in $S_1$. Depending on the particular key agreement mechanism, further conditions can be imposed.

NOTE 4 An example of a possible function FP is given in Annex F.

NOTE 5 As discussed in Clause 6, in practical implementations of the key agreement mechanisms, the shared secret key is subject to further processing. A derived shared secret key is computed by 1) extracting bits from the shared secret key $K_{ABC}$ directly, or 2) passing the shared secret key $K_{ABC}$ and optionally other non-secret data through a one-way function and extracting bits from the output.

### 10.4 Secret key transport

Secret key transport (often abbreviated to "key transport") is the process of transferring a secret key, chosen by one entity (or a trusted centre), to another entity, suitably protected by asymmetric cryptographic encryption.

### 10.5 Public key transport

Public key transport makes an entity's public key available to other entities in an authenticated fashion. Authenticated distribution of public keys is an essential security requirement. This distribution can be achieved in two main ways:

a) public key distribution without a trusted third party;

b) public key distribution involving a trusted third party, such as a certification authority.

The public key of an entity $A$ is part of the public key information of entity $A$. The public key information includes at least entity $A$'s distinguishing identifier and entity $A$'s public key.

## 11 Key agreement

### 11.1 Key agreement mechanism 1

This key agreement mechanism non-interactively establishes a shared secret key between entities $A$ and $B$ with mutual implicit key authentication. The following requirements shall be satisfied.

— Each entity $X$ has a private key agreement key $h_X$ in $S_1$ and a public key agreement key $p_X = \mathrm{F}(h_X, g)$.

— Each entity has access to an authenticated copy of the public key agreement key of the other entity. This can be achieved using the mechanisms described in Clause 13.

Key agreement mechanism 1 is summarized in Figure 1.

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

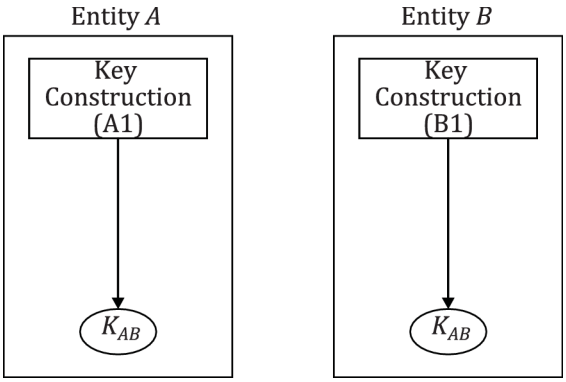


**Figure 1 — Key agreement mechanism 1**

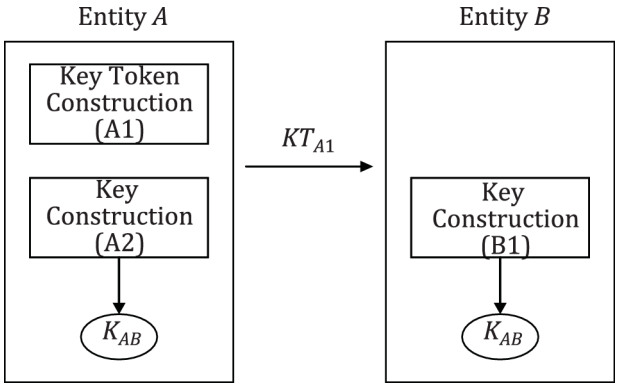


**Figure 2 — Key agreement mechanisms 2 and 8**

**Key construction (A1)** Entity $A$ computes, using its own private key agreement key $h_A$ and entity $B$'s public key agreement key $p_B$, the shared secret key as $K_{AB} = \mathrm{F}(h_A, p_B)$.

**Key construction (B1)** Entity $B$ computes, using its own private key agreement key $h_B$ and entity $A$'s public key agreement key $p_A$, the shared secret key as $K_{AB} = \mathrm{F}(h_B, p_A)$.

As a consequence of requirements on F specified in Clause 10, the two computed values for the key $K_{AB}$ are identical.

NOTE 1 The number of passes is 0.

NOTE 2 This mechanism provides mutual implicit key authentication. However, a zero-pass protocol such as this always generates the same key. One way to eliminate this problem is to ensure that the key is only used once. Furthermore, the use of a unique initialization vector with each utilization of the key can also solve this problem.

NOTE 3 This mechanism does not provide key confirmation.

NOTE 4 This mechanism is a key agreement mechanism, since the established key is a one-way function of the private key agreement keys $h_A$ and $h_B$ of entities $A$ and $B$, respectively. However, one entity can learn the other entity's public key prior to choosing their private key. As described in Clause 8, such an entity can select approximately $s$ bits of the established key, at the cost of generating $2^s$ candidate values for their private key agreement key in the interval between discovering the other entity's public key and choosing their own private key.

NOTE 5 Examples of this mechanism (known as Diffie-Hellman key agreement) are given in Clauses D.2, D.3, and E.3.

NOTE 6 This mechanism has no resilience to key compromise impersonation attacks on $A$.


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

## 11.2 Key agreement mechanism 2

This key agreement mechanism establishes a shared secret key in one pass between entities $A$ and $B$ with implicit key authentication from entity $B$ to entity $A$, but no entity authentication from entity $A$ to entity $B$ (i.e. entity $B$ does not know with whom it has established the shared secret key). The following requirements shall be satisfied.

— Entity $B$ has a private key agreement key $h_B$ in $S_1$ and a public key agreement key $p_B = \mathrm{F}(h_B, g)$.

— Entity $A$ has access to an authenticated copy of entity $B$'s public key agreement key $p_B$. This can be achieved using the mechanisms described in Clause 13.

Key agreement mechanism 2 is summarized in Figure 2.

**Key token construction (A1)** Entity $A$ randomly and secretly generates $r$ in $S_1$, computes $\mathrm{F}(r, g)$ and sends the key token $KT_{A1} = \mathrm{F}(r, g)||Text$ to entity $B$.

**Key construction (A2)** Entity $A$ computes the shared key as $K_{AB} = \mathrm{F}(r, p_B)$.

**Key construction (B1)** Entity $B$ extracts $\mathrm{F}(r,g)$ from the received key token $KT_{A1}$ and computes the shared secret key $K_{AB} = \mathrm{F}(h_B, \mathrm{F}(r, g))$.

As a consequence of the requirements on F specified in Clause 10, the two computed values for the key $K_{AB}$ are identical.

NOTE 1 The number of passes is 1.

NOTE 2 This mechanism provides implicit key authentication from entity $B$ to entity $A$ (entity $B$ is the only entity other than entity $A$ who can compute the shared secret key).

NOTE 3 This mechanism does not provide key confirmation.

NOTE 4 This mechanism is a key agreement mechanism, since the established key is a one-way function of a random value $r$ supplied by entity $A$ and entity $B$'s private key agreement key. As discussed in Clause 8, since entity $A$ can learn entity $B$'s public key prior to choosing the value $r$, entity $A$ can select approximately $s$ bits of the established key, at the cost of generating $2^s$ candidate values for $r$ in the interval between discovering entity $B$'s public key and sending $KT_{A1}$.

NOTE 5 Examples of this mechanism (known as ElGamal key agreement) are described in Clauses D.4 and E.4.

NOTE 6 As entity $B$ receives the information necessary to compute the key $K_{AB}$ from entity $A$, which has not been authenticated, use of $K_{AB}$ by entity $B$ is restricted to functions not requiring trust in entity $A$'s authenticity, such as decryption and generation of message authentication codes.

## 11.3 Key agreement mechanism 3

This key agreement mechanism establishes a shared secret key in one pass between entities $A$ and $B$ with mutual implicit key authentication, and entity authentication of entity $A$ to entity $B$. The following requirements shall be satisfied.

— Entity $A$ has an asymmetric signature system $(\mathrm{S}_A, \mathrm{V}_A)$.

— Entity $B$ has access to an authenticated copy of the public verification transformation $\mathrm{V}_A$. This can be achieved using the mechanisms described in Clause 13.

— Entity $B$ has a key agreement scheme with keys $(h_B, p_B)$.

— Entity $A$ has access to an authenticated copy of the public key agreement key $p_B$ of entity $B$. This can be achieved using the mechanisms described in Clause 13.

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

— (Optional) If used, the *TVP* shall either be a time stamp or a sequence number. If time stamps are used, secure and synchronized time clocks are required; if sequence numbers are used, the ability to maintain and verify bilateral counters is required.

— The entities *A* and *B* have agreed on a MAC function and a way to use $K_{AB}$ as the key for this MAC function. ISO/IEC 9797 is referred for a MAC function.

Key agreement mechanism 3 is summarized in Figure 3.

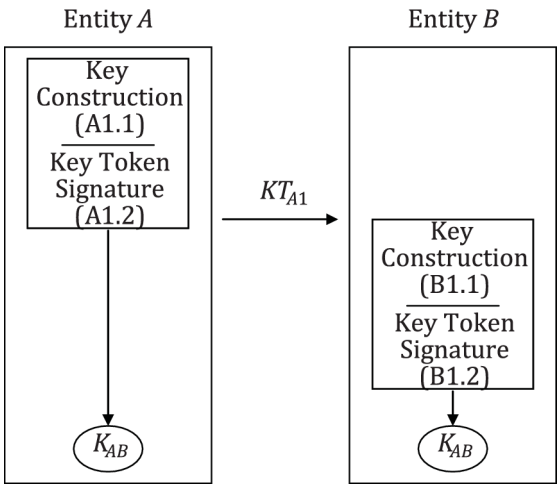


**Figure 3 — Key agreement mechanism 3**

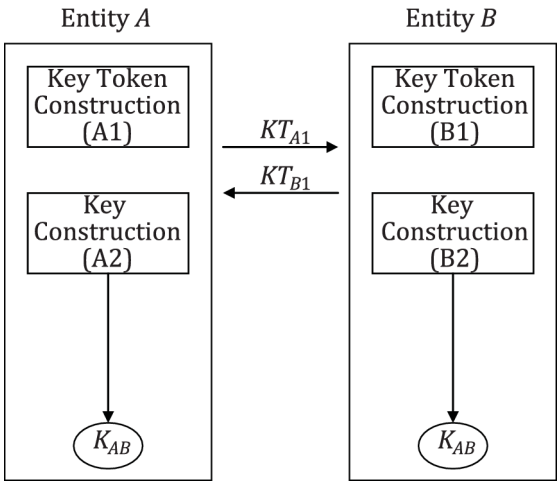


**Figure 4 — Key agreement mechanisms 4, 5 and 9**

**Key construction (A1.1)** Entity *A* randomly and secretly generates *r* in $S_1$ and computes $F(r, g)$. Entity *A* computes the shared secret key as $K_{AB} = F(r, p_B)$.

Using the shared secret key $K_{AB}$, entity *A* computes a MAC on the concatenation of the sender's distinguishing identifier for entity *A* and an optional *TVP*, a time stamp or a sequence number.

**Key token signature (A1.2)** Entity *A* signs the MAC, using its private signature transformation $S_A$. Then entity *A* forms the key token, consisting of the sender's distinguishing identifier for entity *A*, the key input $F(r, g)$, the (optional) *TVP*, the signed MAC, and some optional data, i.e.

$$KT_{A1} = A||F(r,g)||TVP||\ S_A(\mathrm{MAC}_{K_{AB}}(A||TVP))||Text1$$


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

and sends it to entity *B*.

**Key construction (B1.1)** Entity *B* extracts F($r$, $g$) from the received key token and computes the shared secret key, using its private key agreement key $h_B$, $K_{AB}$ = F($h_B$, F($r$, $g$)).

Using the shared secret key $K_{AB}$, entity *B* computes the MAC on the sender's distinguishing identifier for entity *A* and the (optional) *TVP*.

**Signature verification (B1.2)** Entity *B* uses the sender's public verification transformation $V_A$ to verify entity *A*'s signature and, thus, the integrity and origin of the received key token $KT_{A1}$. Then entity *B* validates the timeliness of the token [by inspection of the (optional) *TVP*].

NOTE 1 The number of passes is 1.

NOTE 2 This mechanism provides explicit key authentication from entity *A* to entity *B* and implicit key authentication from entity *B* to entity *A*.

NOTE 3 This mechanism provides key confirmation from entity *A* to entity *B*.

NOTE 4 This mechanism is a key agreement mechanism, since the established key is a one-way function of a random value *r* supplied by entity *A* and entity *B*'s private key agreement key. As discussed in Clause 8, since entity *A* can learn entity *B*'s public key prior to choosing the value *r*, entity *A* can select approximately *s* bits of the established key, at the cost of generating $2^s$ candidate values for *r* in the interval between discovering entity *B*'s public key and sending $KT_{A1}$.

NOTE 5 The (optional) *TVP* prevents replay of the key token from entity *A* to entity *B*.

NOTE 6 Examples of this mechanism (known as Nyberg-Rueppel key agreement) are described in Clauses D.5 and E.5

NOTE 7 If *Text*1 is used to transfer entity *A*'s public key certificate, then requirement 2 at the beginning of 11.3 can be relaxed to the requirement that entity *B* is in possession of an authenticated copy of the CA's public verification key.

NOTE 8 This mechanism has no resilience to key compromise impersonation attacks on *A*.

## 11.4 Key agreement mechanism 4

This key agreement mechanism establishes a shared secret key in two passes between entities *A* and *B* with joint key control without prior exchange of keying information. This mechanism provides neither entity authentication nor key authentication.

Key agreement mechanism 4 is summarized in Figure 4.

**Key token construction (A1)** Entity *A* randomly and secretly generates $r_A$ in $S_1$, computes F($r_A$, $g$), constructs the key token $KT_{A1}$ = F($r_A$, $g$)||*Text*1, and sends it to entity *B*.

**Key token construction (B1)** Entity *B* randomly and secretly generates $r_B$ in $S_1$, computes F($r_B$, $g$), constructs the key token $KT_{B1}$ = F($r_B$, $g$)||*Text*2, and sends it to entity *A*.

**Key construction (A2)** Entity *A* extracts F($r_B$, $g$) from the received key token $KT_{B1}$ and computes the shared secret key $K_{AB}$ = F($r_A$, F($r_B$, $g$)).

**Key construction (B2)** Entity *B* extracts F($r_A$, $g$) from the received key token $KT_{A1}$ and computes the shared secret key $K_{AB}$ = F($r_B$, F($r_A$, $g$)).

NOTE 1 The number of passes is 2.

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

NOTE 2 This mechanism does not provide implicit or explicit key authentication. However, this mechanism can be useful in environments where authenticity of the key tokens is verified using other means. For instance, a hash-code of the key tokens can be exchanged between the entities using a second communication channel. See also public key transport mechanism 2. Another example of entity authentication is using mechanisms specified in Reference [6].

NOTE 3 A separate channel or means exists whereby the key tokens can be verified.

NOTE 4 This mechanism provides no key confirmation.

NOTE 5 This mechanism is a key agreement mechanism, since the established key is a one-way function of random values $r_A$ and $r_B$ supplied by entities $A$ and $B$ respectively. As discussed in Clause 8, since entity $B$ can learn $F(r_A, g)$ prior to choosing the value $r_B$, entity $B$ can select approximately $s$ bits of the established key, at the cost of generating $2^s$ candidate values for $r_B$ in the interval between receiving $KT_{A1}$ and sending $KT_{B1}$.

NOTE 6 Examples of this mechanism (known as Diffie-Hellman key agreement) are described in Clauses D.6 and E.7.

## 11.5 Key agreement mechanism 5

This key agreement mechanism establishes a shared secret key in two passes between entities $A$ and $B$ with mutual implicit key authentication and joint key control. The following requirements shall be satisfied.

— Each entity $X$ has a private key agreement key $h_X$ in $S_1$ and a public key agreement key $p_X = F(h_X, g)$.

— Each entity has access to an authenticated copy of the public key agreement key of the other entity. This can be achieved using the mechanisms described in Clause 13.

— Both entities have agreed on a common one-way function w.

Key agreement mechanism 5 is summarized in Figure 4.

**Key token construction (A1)** Entity $A$ randomly and secretly generates $r_A$ in $S_1$, computes $F(r_A, g)$ and sends the key token $KT_{A1} = F(r_A, g)||Text1$ to entity $B$.

**Key token construction (B1)** Entity $B$ randomly and secretly generates $r_B$ in $S_1$, computes $F(r_B, g)$ and sends the key token $KT_{B1} = F(r_B, g)||Text2$ to entity $A$.

**Key construction (B2)** Entity $B$ extracts $F(r_A, g)$ from the received key token $KT_{A1}$ and computes the shared secret key as $K_{AB} = w(F(h_B, F(r_A, g))||F(r_B, p_A))$ where w is a one-way function.

**Key construction (A2)** Entity $A$ extracts $F(r_B, g)$ from the received key token $KT_{B1}$ and computes the shared secret key as $K_{AB} = w(F(r_A, p_B)||F(h_A, F(r_B, g)))$.

NOTE 1 The number of passes is 2.

NOTE 2 This mechanism provides mutual implicit key authentication. If the data field *Text*2 contains a MAC (on known data) computed using the key $K_{AB}$, then this mechanism provides explicit key authentication from entity $B$ to entity $A$.

NOTE 3 If the data field *Text*2 contains a MAC (on known data) computed using the key $K_{AB}$, then this mechanism provides key confirmation from entity $B$ to entity $A$.

NOTE 4 This mechanism is a key agreement mechanism, since the established key is a one-way function of random values $r_A$ and $r_B$ supplied by entities $A$ and $B$ respectively.

NOTE 5 Examples of this key agreement mechanism [known as the Matsumoto-Takashima-Imai A(0) key agreement scheme] are described in Clauses D.7 and E.6. Another example is known as the Goss protocol.


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

NOTE 6 If *Text*1 and *Text*2 contain the public key certificates of entity *A*'s and *B*'s key agreement keys, respectively, then the requirement 2 at the beginning of 11.5 can be replaced by the requirement that each entity is in possession of an authenticated copy of the CA's public verification key.

NOTE 7 Under certain circumstances, this mechanism can be subject to a source substitution attack (also known as an unknown key share attack)[30]. If this is a concern, this type of attack can be avoided by ensuring that as part of the process of submitting a public key to a CA for certification, the submitter proves possession of the corresponding private key. This type of attack is slightly more serious in the case of protocols based on elliptic curves.[26] Resilience to unknown key share attack for *A* and *B* can be achieved by choosing a key derivation function that includes the identifiers of the involved entities.

## 11.6 Key agreement mechanism 6

This key agreement mechanism establishes a shared secret key in two passes between entities *A* and *B* with mutual implicit key authentication and joint key control. It is based on the use of both an asymmetric encryption scheme and a signature system. The following requirements shall be satisfied.

a) Entity *A* has an asymmetric encryption system with transformations ($E_A$, $D_A$).

b) Entity *B* has an asymmetric signature system with transformations ($S_B$, $V_B$).

c) Entity *A* has access to an authenticated copy of entity *B*'s public verification transformation $V_B$. This can be achieved using the mechanisms described in Clause 13.

d) Entity *B* has access to an authenticated copy of entity *A*'s public encryption transformation $E_A$. This can be achieved using the mechanisms described in Clause 13.

Key agreement mechanism 6 is summarized in Figure 5.

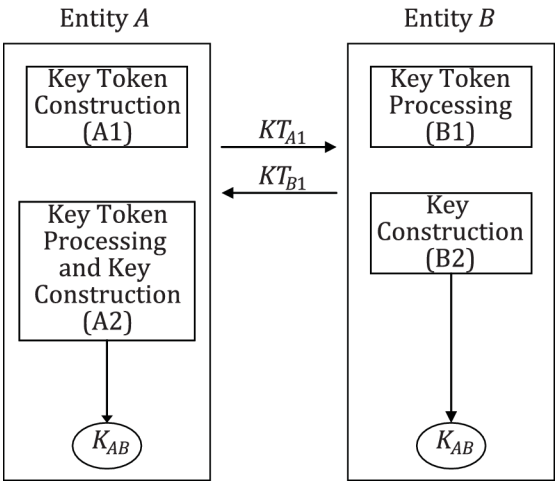


**Figure 5 — Key agreement mechanism 6**

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

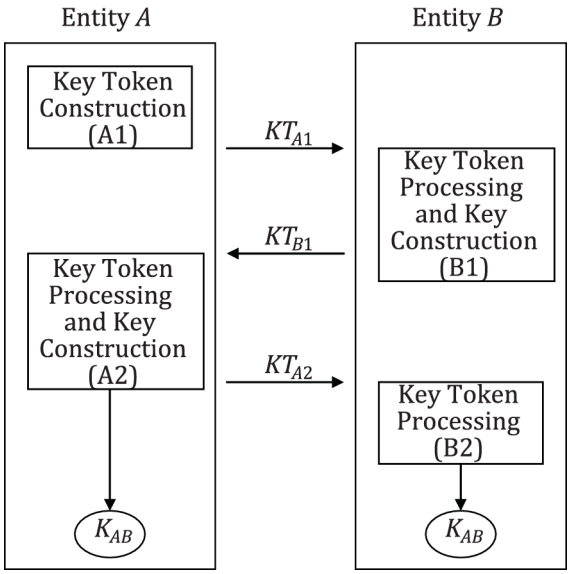


**Figure 6 — Key agreement mechanism 7**

**Key token construction (A1)** Entity $A$ generates a random number $r_A$, constructs the key token $KT_{A1}$ = $r_A||Text1$, and sends it to entity $B$.

**Key token processing (B1)** Entity $B$ generates a random number $r_B$ and signs a data block consisting of the distinguishing identifier for entity $A$, the random number $r_A$, the random number $r_B$ and some optional data $Text2$ using its private signature transformation $S_B$, to obtain $BS = S_B(A||r_A||r_B||Text2)$.

Entity $B$ then encrypts a data block consisting of its distinguishing identifier (optional), the signed block $BS$, and some optional data $Text3$ using entity $A$'s public encryption transformation $E_A$. Entity $B$ then sends the key token $KT_{B1} = E_A(BS||Text3)||Text4$ back to entity $A$, or entity $B$ can include the identifier for $B$ as $KT_{B1}= E_A(B||BS||Text3)\ ||Text4$.

**Key construction (B2)** The shared secret key consists of all or part of entity $B$'s signature $\Sigma$ contained in the signed block $BS$ (see Note 2 in Clause 4), after passing through a key derivation function.

**Key token processing and key construction (A2)** Entity $A$ decrypts the key token $KT_{B1}$ using its private decryption transformation $D_A$, optionally checks the sender identifier, and uses entity $B$'s public verification transformation $V_B$ to verify the digital signature of the signed block $BS$. Then entity $A$ checks the recipient identifier and that the random number $r_A$ in the signed block $BS$ equals the random number $r_A$ sent in token $KT_{A1}$. If all checks are successful, entity $A$ accepts all or part of entity $B$'s signature of the signed block $BS$ used with a key derivation function as the shared secret key.

NOTE 1 The number of passes is 2.

NOTE 2 The part of the signature $\Sigma$ that is to be used as the basis of the secret key established between entities $A$ and $B$ is agreed in advance.

NOTE 3 This mechanism provides implicit key authentication from entity $A$ to entity $B$ and explicit key authentication from entity $B$ to entity $A$.

NOTE 4 If the data field $Text3$ contains a MAC (on known data) computed using the key $K_{AB}$, then this mechanism provides key confirmation from entity $B$ to entity $A$.

NOTE 5 This mechanism is a key agreement mechanism, since the established key is a one-way function of random values $r_A$ and $r_B$ supplied by entities $A$ and $B$ respectively. As discussed in Clause 8, since entity $B$ can learn $F(r_A, g)$ prior to choosing the value $r_B$, entity $B$ can select approximately $s$ bits of the established key, at the cost of generating $2^s$ candidate values for $r_B$ in the interval between receiving $KT_{A1}$ and sending $KT_{B1}$.


© ISO/IEC 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021

NOTE 6 This mechanism is derived from Beller and Yacobi's two pass protocol described in Clause D.8.

NOTE 7 If *Text*1 and *Text*4 contain a public key certificate for entity *A*'s encryption key and a public key certificate for entity *B*'s verification key, respectively, then the requirements 3 and 4 at the beginning of 11.6 can be relaxed to the requirement that each entity is in possession of an authenticated copy of the CA's public verification key.

NOTE 8 A significant feature of this scheme is that the identity of entity *B* can remain anonymous to eavesdroppers, a property of potential significance in a wireless communication environment.

NOTE 9 This mechanism has no resilience to key compromise impersonation attacks on *A*.

## 11.7 Key agreement mechanism 7

This key agreement mechanism is based on the three-pass authentication mechanism of ISO/IEC 9798-3 and establishes a shared secret key between entities *A* and *B* with mutual authentication. The following requirements shall be satisfied.

— Each entity *X* has an asymmetric signature system $(S_X, V_X)$.

— Both entities have access to an authenticated copy of the public verification transformation of the other entity. This can be achieved using the mechanisms described in Clause 13.

— The two entities have agreed on a common MAC function.

Key agreement mechanism 7 is summarized in Figure 6.

**Key token construction (A1)** Entity *A* randomly and secretly generates $r_A$ in $S_1$, computes $F(r_A, g)$, constructs the key token $KT_{A1} = F(r_A, g)||Text1$, and sends it to entity *B*.

**Key token processing and key construction (B1)** Entity *B* randomly and secretly generates $r_B$ in $S_1$, computes $F(r_B, g)$, computes the shared secret key as $K_{AB} = F(r_B, F(r_A, g))$, and constructs the signed key token

$$KT_{B1} = S_B(DB_1)||\,MAC_{K_{AB}}(DB_1)||Text3,$$

where $DB_1 = F(r_B, g)||F(r_A, g)||A||Text2$, and sends it back to entity *A*.

Key confirmation is provided by including $MAC_{K_{AB}}(DB_1)$ in $KT_{B1}$. Alternatively, if both parties have a common symmetric encryption system, key confirmation can be obtained by replacing $KT_{B1}$ with $KT_{B1} = F(r_B, g)||\,E_{K_{AB}}(S_B(DB_1))$, where E is a suitable symmetric encryption function.

**Key token processing and key construction (A2)** Entity *A* verifies entity *B*'s signature on the key token $KT_{B1}$ using entity *B*'s public verification key, and then verifies entity *A*'s distinguishing identifier and the value $F(r_A, g)$ sent in step (A1). If the checks are successful, entity *A* proceeds to compute the shared secret key as $K_{AB} = F(r_A, F(r_B, g))$.

Using $K_{AB}$, entity *A* verifies $MAC_{K_{AB}}(DB_1)$. Then entity *A* constructs the signed key token

$$KT_{A2} = S_A(DB_2)||\,MAC_{K_{AB}}(DB_2)||Text5,$$

where $DB_2 = F(r_A, g)||F(r_B, g)||B||Text4$, and sends it to entity *B*.

Key confirmation is provided by including $MAC_{K_{AB}}(DB_2)$ in $KT_{A2}$. Alternatively, key confirmation can be obtained by replacing $KT_{A2}$ with $KT_{A2} = E_{K_{AB}}(S_A(DB_2))$.

© ISO/IEC 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 11770-3:2021