Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com/. Reproduction and distribution is prohibited.

BS ISO/IEC 9798-6:2010

BSI Standards Publication

Information technology — Security techniques — Entity authentication

Part 6: Mechanisms using manual data transfer

bsi.

...making excellence a habit.™

National foreword

This British Standard is the UK implementation of ISO/IEC 9798-6:2010. It supersedes BS ISO/IEC 9798-6:2005 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/33, IT - Security techniques.

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

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© BSI 2010

ISBN 978 0 580 68073 1

ICS 35.040

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 December 2010.

Amendments issued since publication

Date	Text affected

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

BS ISO/IEC 9798-6:2010

INTERNATIONAL STANDARD

ISO/IEC 9798-6

Second edition
2010-12-01

Information technology — Security techniques — Entity authentication —

Part 6: Mechanisms using manual data transfer

Technologies de l'information — Techniques de sécurité — Authentification d'entité —

Partie 6: Mécanismes utilisant un transfert manuel de données

Reference number
ISO/IEC 9798-6:2010(E)

© ISO/IEC 2010

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2010

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Contents

Page

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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.

ISO/IEC 9798-6 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

This second edition cancels and replaces the first edition (ISO/IEC 9798-6:2005), to which a new Clause 7 has been added to provide four new mechanisms. It also incorporates the Technical Corrigendum ISO/IEC 9798-6:2005/Cor.1:2009. Implementations conformant to the first edition will be conformant to the second edition.

ISO/IEC 9798 consists of the following parts, under the general title *Information technology — Security techniques — Entity authentication*:

— *Part 1: General*

— *Part 2: Mechanisms using symmetric encipherment algorithms*

— *Part 3: Mechanisms using digital signature techniques*

— *Part 4: Mechanisms using a cryptographic check function*

— *Part 5*: *Mechanisms using zero-knowledge techniques*

— *Part 6: Mechanisms using manual data transfer*

© ISO/IEC 2010 – All rights reserved

Introduction

Within networks of communicating devices it is often necessary for two devices to perform an entity authentication procedure using a channel which may be subject to both passive and active attacks, where an active attack can include a malicious third party introducing data into the channel and/or modifying, deleting or repeating data legitimately sent on the channel. Other parts of ISO/IEC 9798 specify entity authentication mechanisms applicable when the two devices share a secret key, or where one device has an authenticated copy of a public key for the other device.

In this part of ISO/IEC 9798, entity authentication mechanisms where there is no such assumption of pre-established keying relationships, referred to as manual authentication mechanisms, are specified. Instead entity authentication is achieved by manually transferring short data strings from one device to the other, or by manually comparing short data strings output by the two devices.

For the purposes of this part of ISO/IEC 9798, the meaning of the term entity authentication is different from the meaning applied in other parts of ISO/IEC 9798. Instead of one device verifying that the other device has a claimed identity (and vice versa), both devices in the possession of a user verify that they correctly share a data string with the other device at the time of execution of the mechanism. Of course, this data string could contain identifiers for one or both of the devices.

As described in informative Annexes B and C, a manual authentication mechanism can be used as the basis for secret key establishment or the reliable exchange of public keys. A manual authentication mechanism could also be used for the reliable exchange of other secret or public security parameters, including security policy statements or timestamps.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

INTERNATIONAL STANDARD **ISO/IEC 9798-6:2010(E)**

Information technology — Security techniques — Entity authentication —

Part 6: Mechanisms using manual data transfer

1 Scope

This part of ISO/IEC 9798 specifies eight entity authentication mechanisms based on manual data transfer between authenticating devices. It indicates how these mechanisms can be used to support key management functions, and provides guidance on secure choices of parameters for the mechanisms. A comparison of the levels of security and efficiency provided by the eight mechanisms is given.

Such mechanisms can be appropriate in a variety of circumstances. One such application occurs in personal networks, where the owner of two personal devices capable of wireless communications wishes them to perform an entity authentication procedure as part of the process of preparing them for use in the network.

2 Normative references

The following referenced documents are indispensable for the application 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 9798-1:2010, *Information technology — Security techniques — Entity authentication — Part 1: General*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 9798-1 and the following apply.

3.1
check-value
string of bits, computed as the output of a check-value function, sent from the data originator to the data recipient that enables the recipient of data to check its correctness

3.2
check-value function
function f which maps a string of bits and a short secret key, i.e. a key that can readily be entered into or read from a user device, to a fixed-length string of bits, i.e. a b-bit check-value, satisfying the following properties:

— for any key k and any input string d, the function $f(d, k)$ can be computed efficiently;

— it is computationally infeasible to find a pair of distinct data strings (d, d') for which the number of keys which satisfy $f(d, k) = f(d', k)$ is more than a small fraction of the possible set of keys.

NOTE In practice, a short key would typically contain 4–6 digits or alphanumeric characters.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

3.3
data origin authentication
corroboration that the source of data received is as claimed

[ISO 7498-2]

3.4
digest-value
string of bits, computed as the output of a digest function, sent from the data originator to the data recipient that enables the recipient of data to check its correctness

3.5
digest function
function d which maps a string of bits and a long secret key to a short and fixed-length string of bits, i.e. a b-bit digest-value, that can readily be entered into or read from a user device, satisfying the following properties:

— for any key k and any input string m, the function $d(m, k)$ can be computed efficiently;

— it is computationally infeasible to find a pair of distinct data strings (m, m') for which the proportion of keys which satisfy $d(m, k) = d(m', k)$ is greater than $(2^{-b} + \varepsilon)$, where b is the bit length of a digest-value and ε is a value that is negligible relative to 2^{-b}.

NOTE 1 In practice, the second digest function property should be satisfied if the key k is of the size of a typical cryptographic hash value, for example, 160 bits. This requirement derives from theoretical lower bounds on the key length for universal hash-functions, which are a general class of digest functions. More detailed discussions of this issue can be found in Annex F.

NOTE 2 See Annexes D, F, and G for further discussions of key and digest lengths.

3.6
hash-function
function which maps strings of bits to fixed-length strings of bits, satisfying the following two properties:

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

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

[ISO/IEC 10118-1]

3.7
manual authentication certificate
combination of a secret key and a check-value, generated by one of the two devices engaging in manual authentication, with the property that, when entered into the other device, this pair of values can be used to complete the manual authentication process at some later time

3.8
Message Authentication Code
MAC
string of bits which is the output of a MAC algorithm

[ISO/IEC 9797-1]

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

3.9
Message Authentication Code algorithm
MAC algorithm
algorithm for computing a function which maps strings of bits and a secret key to fixed-length strings of bits, satisfying the following properties:

— for any key and any input string the function can be computed efficiently;

— for any fixed key, and given no prior knowledge of the key, it is computationally infeasible to compute the function value on any new input string, even given knowledge of the set of input strings and corresponding function values, where the value of the ith input string may have been chosen after observing the value of the first i-1 function values.

[ISO/IEC 9797-1]

3.10
manual entity authentication
process achieving entity authentication between two devices using a combination of message exchanges via a (potentially insecure) communications channel and the manual transfer of limited amounts of data between the devices

3.11
simple input interface
interface for a device that allows the user to indicate to the device the successful or unsuccessful completion of a procedure, e.g. as could be implemented as a pair of buttons or a single button which is either pressed or not within a certain time interval

3.12
simple output interface
interface for a device that allows the device to indicate to the user the successful or unsuccessful completion of a procedure, e.g. as could be implemented by red and green lights or as single light which is lit in different ways to indicate success or failure

4 Symbols and abbreviated terms

A, B	Labels used for the two devices engaging in a manual entity authentication mechanism
d	Digest function, used in mechanisms 3 and 5, where $d(D,k)$ denotes the digest value computed on data string D using key k
D	Data string whose value is established between devices A and B as the result of performing a manual entity authentication mechanism
h	Hash-function, used in mechanisms 3–6
I_A, I_B	Distinguishing identifiers of A and B respectively
K	(Short) secret key used with a check-value function in mechanisms 1 and 2
k	(Long) secret key used in mechanisms 3–6
K_A, K_{Ai}, K_B, K_{Bi}	Random MAC keys used in mechanisms 7 and 8
MAC	Message Authentication Code
R	(Short) random bit-string used in mechanisms 4, 6, 7, and 8

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

|| As defined in ISO/IEC 9798-1, $X||Y$ is used to mean the result of the concatenation of data items X and Y in the order specified. In cases where the result of concatenating two or more data items is input to a function as part of one of the mechanisms specified in this document, this result shall be composed so that it can be uniquely resolved into its constituent data strings, i.e. so that there is no possibility of ambiguity in interpretation. This latter property could be achieved in a variety of different ways, depending on the application. For example, it could be guaranteed by (a) fixing the length of each of the substrings throughout the domain of use of the mechanism, or (b) encoding the sequence of concatenated strings using a method that guarantees unique decoding, e.g. using the distinguished encoding rules defined in ISO/IEC 8825-1 [10]

NOTE Annexes D and F provide guidance on appropriate choices for the lengths of short secrets and MAC keys.

5 Overall requirements

The authentication mechanisms specified in this document have the following requirements in addition to the specific requirements specified in Clauses 6, 7 and 8.

a) The pair of devices performing the manual authentication procedure shall be connected via a communications link (e.g. a wireless or Internet link). No security assumptions are made regarding this link; that is, the mechanisms are designed to operate securely even in an environment where an attacker can monitor and change data transferred on this link.

b) The pair of devices performing the manual authentication procedure shall both have a user interface capable of data input and data output.

c) The user data input interface for a device shall, at minimum, be capable of indicating successful or unsuccessful completion of a procedure (e.g. as could be implemented by using either two buttons or a single button which is either pressed or not within a certain time interval); such a means of data input is referred to below as a *simple* input interface. By contrast, a *standard* input interface shall provide means for the input of a short string of symbols, e.g. a numeric, hexadecimal, or alphanumeric keypad. Unless explicitly stated otherwise, it is necessary that every device has a standard means of data input.

d) The user data output interface for a device shall, at minimum, be capable of indicating either success or failure of an authentication procedure (e.g. as could be implemented by means of red and green lights); such a means of data output is referred to below as a *simple* output interface. By contrast, a *standard* output interface shall provide means for the output of a short string of symbols, e.g. a numeric, hexadecimal or alphanumeric display. Unless explicitly stated otherwise, it is necessary that every device has a standard means of data output.

e) For mechanisms 1 and 2, the two devices performing the entity authentication procedure shall have agreed on the use of a specific check-value function, and shall have the means to implement this function.

NOTE 1 Guidance on appropriate choices for check-value functions and lengths for check-values and random keys for use in mechanisms 1 and 2 is provided in Annex D. A construction for an unconditionally secure check-value function suitable for use with mechanisms 1 and 2 is given in Annex E.

f) For mechanisms 3–6, the two devices performing the entity authentication procedure shall have agreed on the use of a specific hash-function h, and shall have the means to implement this function.

NOTE 2 Guidance on appropriate choices for bit lengths for hash-function inputs and outputs for use in mechanisms 3–6 is provided in Annex D.

g) For mechanisms 3 and 5, the two devices performing the entity authentication procedure shall have agreed on the use of a specific digest function d, and shall have the means to implement this function.

NOTE 3 Guidance on digests lengths for use in mechanisms 3 and 5 is provided in Annex D. Constructions for digest functions using MAC algorithms and hash-functions, which are suitable for use with mechanisms 3 and 5, are given in Annex G.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

h) For mechanisms 7 and 8, the two devices performing the entity authentication procedure shall have agreed on the use of a specific MAC algorithm, and shall have the means to implement this algorithm.

NOTE 4 Guidance on appropriate choices for MAC algorithms and lengths for MACs and random keys for use in mechanisms 7 and 8 is provided in Annex D.

i) Prior to invocation of mechanisms 1–8, the two devices performing the mechanism shall have exchanged a data string D (in combination with a hash-value in mechanisms 3–6). D may be generated by one device and sent to the other device, or it may consist of the concatenation of data generated by both devices and sent in both directions across the shared communications link.

j) Either a single human user shall be in possession of both devices and shall operate them both, or the two devices shall be operated by two users who share a trusted means of communication.

k) The users of the devices shall be present through the complete running operation to ensure correct processing of these mechanisms. There shall not be a significant delay in manual transfer of data between the devices during this operation. The devices shall automatically time out as indicated in the mechanism specifications to preclude certain attacks.

6 Mechanisms using a short check-value

6.1 General

In this clause two manual authentication mechanisms are specified that are based on the use of a check-value. The two mechanisms are appropriate for different types of devices. Specifically,

— the first mechanism (mechanism 1) is appropriate for the case where one device has a simple input interface and the other has a simple output interface, and

— the second mechanism (mechanism 2) is appropriate for the case where both devices have a simple input interface.

A standard input or output interface can emulate a simple interface, and hence if both devices have standard input and output interfaces then either of the mechanisms may be used.

Both mechanisms operate in the following general way. A data string D is transferred from one device to the other (or is the concatenation of data transferred in both directions) via the shared communications link. The manual entity authentication mechanism is then executed. As a result of the mechanism both devices are provided with assurance that the data string D they possess is the same as the value held by the other device.

6.2 Mechanism 1 – One device with simple input, one device with simple output

6.2.1 Specific requirements

This mechanism has the following specific requirements.

a) The mechanism specified in this subclause is appropriate for the case where one device (device A) has a simple input interface and the other (device B) has a simple output interface.

b) Device A shall have the means to generate keys.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

6.2.2 Specification of data exchanged

The following data exchanges and operations shall take place (see also Figure 1).

a) Both devices shall output a signal to acknowledge that they have received data D and that they are ready for the authentication mechanism to commence. On observing that both devices are ready, the user shall then enter a signal into device A to notify A that the mechanism can start.

b) Device A shall generate a random key K, where K is suitable for use with the check-value function shared by the two components. Using this key K, device A shall compute a check-value as a function of the data D. The check-value and the key K shall then be output via the output interface of device A. The user shall read the check-value and the key K from the output interface.

c) The user shall enter the check-value and the key K output by device A to device B using its input interface. Device B shall use the key K to re-compute the check-value as a function of its stored version of data D. If the two check-values agree, then device B shall output a success signal to the user via its simple output interface. Otherwise it shall give a failure signal.

d) The user shall enter the result output by device B, i.e. success or failure, into device A via its simple input interface.

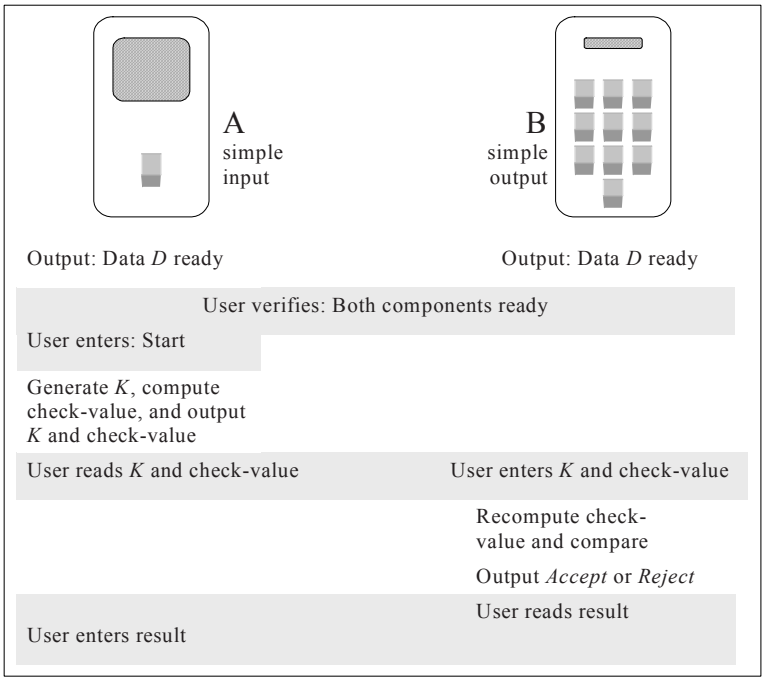

Figure 1 — Manual authentication mechanism 1

6.2.3 Manual authentication certificates

Manual authentication mechanism 1 has the property that no authentication information is transmitted over the insecure channel. Therefore, it does not affect the security of the mechanism if the manual authentication values K and check-value are transferred from device A to device B before the latter has received the actual data D. Naturally, such an approach is applicable only to situations where device A generates the data D. However, in such a case, mechanism 1 offers a means of authenticating data to be received at some later time. Such authentication means is called a manual authentication certificate. A protocol for data origin authentication using a manual authentication certificate is now specified (with the same requirements as specified in subclause 6.2.1). Note that this protocol does not provide entity authentication.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Suppose device A has data D that needs to be sent to device B at some later time.

a) Device A generates a random key K, where K is suitable for use with the check-value function shared by the two devices. Using this key K, device A computes a check-value as a function of the data D. The check-value and the key K are then output to the user by the output interface of device A. The user reads the output check-value and key K.

b) The user enters the check-value and key K output from device A to the input interface of device B. The key K and the check-value are stored in device B.

c) When device B at some later time receives data D, it can verify the authenticity of the data using the stored values of K and the check-value. Device B uses the key K to recompute the check-value as a function of the received data D. If the two check-values agree then device B accepts the data and outputs a success signal to the user. Otherwise it gives a failure signal.

The manual authentication certificate consists of K and the check-value computed as a function of K and D.

NOTE An example of data that could be included in D are a public key of a device, its identity, the domain of service, etc. In Annex B an example is provided of how manual authentication certificates can be used to establish a shared secret key between two devices.

6.3 Mechanism 2 – Devices with simple input capabilities

6.3.1 Specific requirements

This mechanism has the following specific requirements.

a) The mechanism specified in this subclause is appropriate for the case where both devices (A and B) have a simple input interface.

b) One of the devices (the device labelled A below) shall have the means to generate keys.

6.3.2 Specification of data exchanged

The following data exchanges and operations shall take place (see also Figure 2).

a) Both devices shall output a signal to acknowledge that they have received data D and that they are ready for the authentication mechanism to commence. On observing that both devices are ready, the user shall then enter a signal into device A to notify A that the mechanism can start.

b) Device A shall generate a random key K, where K is suitable for use with the check-value function shared by the two components. Using this key K, device A shall compute a check-value as a function of the data D. The check-value and the key K shall then be output via the output interface of device A. Device A shall also transmit the key K to device B via the shared communications link.

c) Device B shall use the key K to compute the check-value as a function of its stored version of data D, and shall output the key K and the computed check-value.

d) The user shall compare the two output check-values and the two output keys. If the values agree then the user enters a signal of acceptance into both devices. If the check-values **or** the key values are different then the mechanism has failed and the user shall enter a rejection signal into the devices. The devices shall interpret the absence of an acceptance signal as a failure signal (this will require the implementation of a time-out mechanism).

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

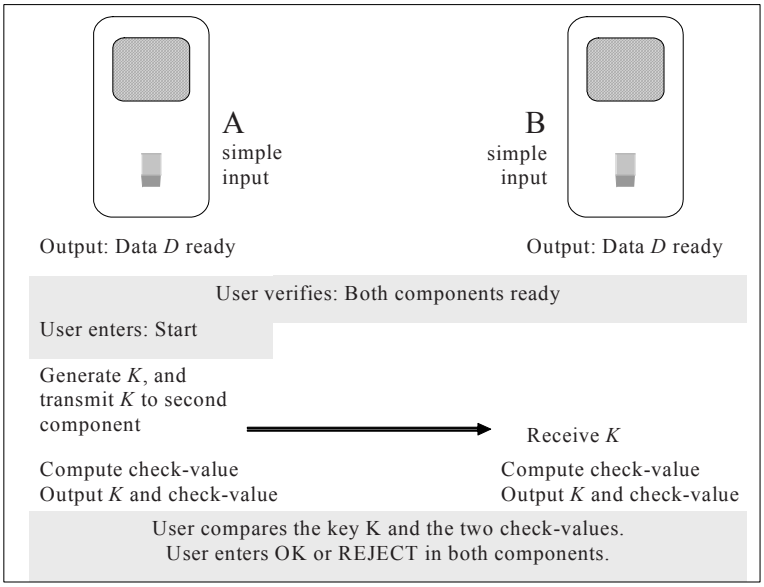

Figure 2 — Manual authentication mechanism 2

7 Mechanisms using a manual transfer of a short digest-value or a short key

7.1 General

In this clause four manual authentication mechanisms are specified that involve the manual transfer of either a short digest-value or a short key. The four mechanisms are appropriate for different types of device. Specifically,

— the first two mechanisms (mechanisms 3 and 4) are appropriate for the case where one device has a simple input interface and the other has a simple output interface, and

— the second two mechanisms (mechanisms 5 and 6) are appropriate for the case where both devices have a simple input interface.

A standard input or output interface can emulate a simple interface, and hence if both devices have standard input and output interfaces then either of the mechanisms can be used.

All mechanisms operate in the following general way. A data string D and a hash-value are transferred from one device to the other (D may alternatively be made up of the concatenation of data transferred in both directions) via the shared communications link. The manual entity authentication mechanism is then executed. As a result of the mechanisms both devices are provided with assurance that the data string D they possess is the same as the value held by the other device.

7.2 Mechanism 3 – One device with simple input, one device with simple output

7.2.1 Specific requirements

This mechanism has the following specific requirements.

a) The mechanism specified in this subclause is appropriate for the case where one device (device A) has a simple input interface and the other (device B) has a simple output interface.

b) Device A shall have the means to generate (long) random keys.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

7.2.2 Specification of data exchanged

The following data exchanges and operations shall take place (see also Figure 3). Note that steps a)–b) may occur in parallel, as may steps d)–e).

a) Devices A and B shall provisionally agree on a data string D. This could, for example, be achieved via an exchange of unprotected messages sent via the shared communications link.

b) Device A shall generate and keep secret a random key k, where k shall be suitable for use as a key with the agreed digest function d. Device A shall compute $h(k)$ and then pass this hash-value to device B by some (not necessarily secure) means, e.g. via the shared communications link.

c) Both devices shall output a signal via their output interfaces to acknowledge that they have completed steps a)–b), and that they are ready for the authentication mechanism to commence. On observing the signals, the user shall enter a signal into device A via its simple input interface to notify A that the mechanism can start.

d) Device A shall compute a short digest-value $d(D, k)$ and output the digest-value via its standard output interface. The user shall read the short digest-value from the standard output interface of A, and enter it into B using B's standard input interface.

e) Device A shall transmit the key k to device B via the shared communications link. On receipt of k, device B shall compute $h(k)$ and check whether it equals the value device B received from A in step b). If the hash-values agree then B proceeds to step f). Otherwise B shall give a failure signal, and shall implement a mechanism whereby it will refuse to start a new instance of the mechanism for a short time-period.

f) Device B shall use the key k and its stored version of data D to re-compute the short digest-value $d(D, k)$. If this digest-value equals the value device B received in step d) then B shall output a success signal to the user via its simple output interface. Otherwise it shall give a failure signal, and shall implement a mechanism whereby it will refuse to start a new instance of the mechanism for a short time-period.

g) The user shall enter the result output by device B, i.e. success or failure, into device A via its simple input interface. A shall interpret the absence of an acceptance signal as a failure signal (this will require the implementation of a time-out mechanism).

NOTE 1 The time delay mechanism used in steps e) and f) prevents a man-in-the-middle attack in which an attacker tries to masquerade as device A to device B immediately after B abandons the run while A is still waiting for an acceptance signal in step g).

NOTE 2 In this mechanism, device B trusts device A since A generates the random key k and so pre-determines the digest-value $d(D, k)$. If B is given the ability to generate random keys then A and B do not have to trust each other. This, however, potentially makes the mechanism more complex because of the increased network communication and synchronisation requirements. This observation also applies to mechanisms 4–6.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

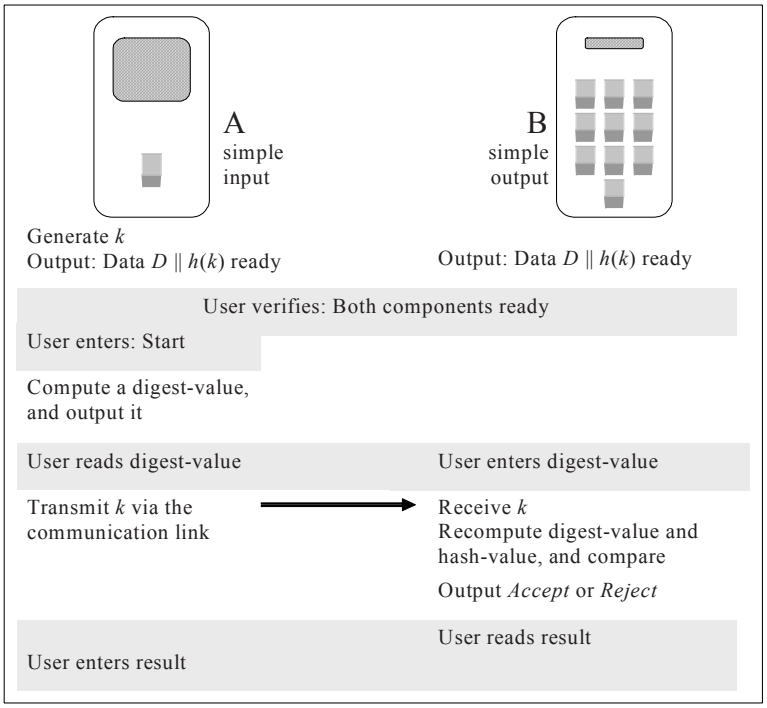

Figure 3 — Manual authentication mechanism 3

7.3 Mechanism 4 – One device with simple input, one device with simple output

7.3.1 Specific requirements

This mechanism has the following specific requirements.

a) The mechanism specified in this subclause is appropriate for the case where one device (device A) has a simple input interface and the other (device B) has a simple output interface.

b) Device A shall have the means to generate (long) random keys and short random bit-streams.

7.3.2 Specification of data exchanged

The following data exchanges and operations shall take place (see also Figure 4). Note that steps a)–b) may occur in parallel, as may steps d)–e).

a) Devices A and B shall provisionally agree on a data string D. This could, for example, be achieved via an exchange of unprotected messages sent via the shared communications link.

b) Device A shall generate and keep secret a (long) random key k and a (short) random bit-stream R. Device A shall compute $h(D||k||R)$ and then pass this hash-value to device B by some (not necessarily secure) means, e.g. via the shared communications link.

c) Both devices shall output a signal via their output interfaces to acknowledge that they have completed steps a)–b), and that they are ready for the authentication mechanism to commence. On observing the signals, the user shall enter a signal into device *A* via its simple input interface to notify A that the mechanism can start.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

d) Device A shall output the short random bit-stream R via its standard output interface. The user shall read the short random bit-stream R from the standard output interface of A, and enter it into B using B's standard input interface.

e) Device A shall transmit the key k to device B via the shared communications link.

f) On receipt of R and k in steps d)–e), device B shall use them to re-compute $h(D||k||R)$ as a function of its stored version of data D. If this hash-value equals the value device B received from A in step b) then B shall output a success signal to the user via its simple output interface. Otherwise it shall output a failure signal.

g) The user shall enter the result output by device B, i.e. success or failure, into device A via its simple input interface. A shall interpret the absence of an acceptance signal as a failure signal (this will require the implementation of a time-out mechanism).

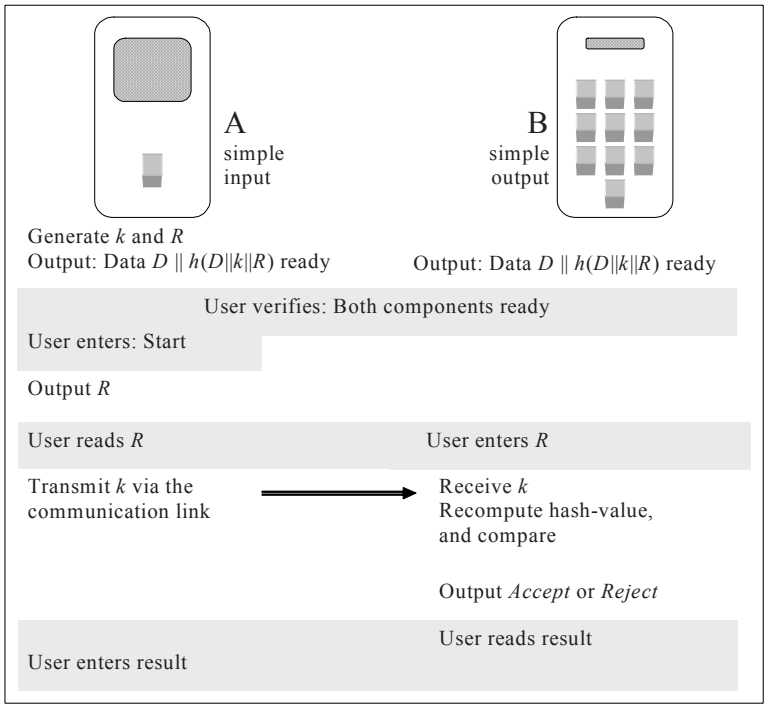

Figure 4 — Manual authentication mechanism 4

7.4 Mechanism 5 – Devices with simple input capabilities

7.4.1 Specific requirements

This mechanism has the following specific requirements.

a) The mechanism specified in this subclause is appropriate for the case where both devices (A and B) have a simple input interface.

b) One of the devices (the device labelled A below) shall have the means to generate (long) random keys.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

7.4.2 Specification of data exchanged

The following data exchanges and operations shall take place (see also Figure 5). Note that steps a)–b) may occur in parallel, as may steps d)–e).

a) Devices A and B shall provisionally agree on a data string D. This could, for example, be achieved via an exchange of unprotected messages sent via the shared communications link.

b) Device A shall generate and keep secret a random key k, where k shall be suitable for use as a key with the agreed digest function d. Device A shall compute $h(k)$ and then pass this hash-value to device B by some (not necessarily secure) means, e.g. via the shared communications link.

c) Both devices shall output a signal via their output interfaces to acknowledge that they have completed steps a)–b), and that they are ready for the authentication mechanism to commence. On observing the signals, the user shall enter a signal into device A via its simple input interface to notify A that the mechanism can start.

d) Device A shall transmit the key k to device B via the shared communications link.

e) Device A shall compute a short digest-value $d(D, k)$ and output the short digest-value via its standard output interface.

f) On receipt of key k in step d), B shall re-compute the hash-value $h(k)$ and the short digest-value $d(D, k)$ as a function of its stored version of D. If this hash-value equals the value device B received from A in step b) then device B shall output the short digest-value via its standard output interface. Otherwise B shall give a failure signal, and shall implement a mechanism whereby it will refuse to start a new instance of the mechanism for a short time-period.

g) The user shall compare the two short digest-values output via the standard output interfaces of devices A and B in steps e)–f). If they are the same then the user shall enter a signal of acceptance into both devices via their simple input interfaces. Otherwise the mechanism has failed and the user shall enter a rejection signal into the devices. The devices shall interpret the absence of an acceptance signal as a failure signal; this will require the implementation of a time-out mechanism.

NOTE The time delay mechanism used in step f) prevents a man-in-the-middle attack in which an attacker tries to masquerade as device A to device B immediately after B abandons the run while A is still waiting for an acceptance signal in step g).

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

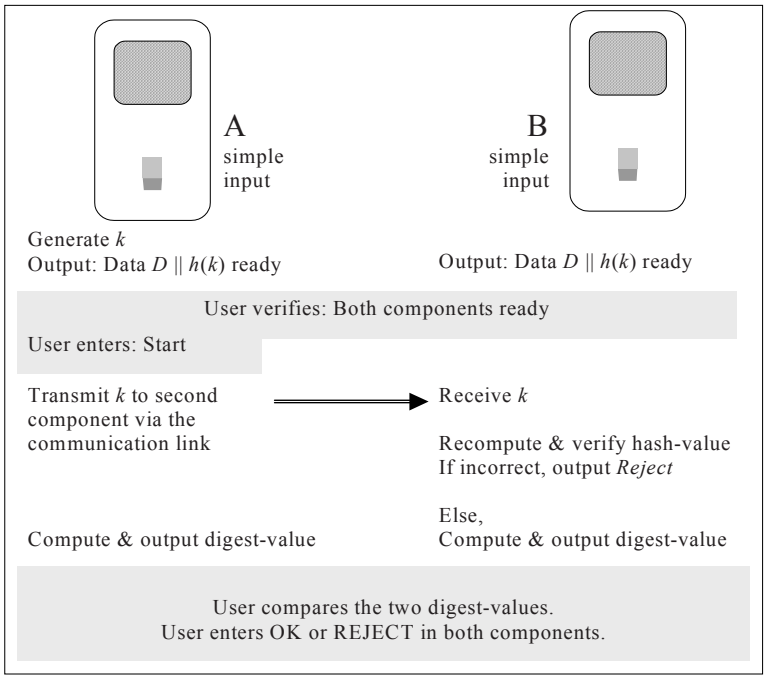

Figure 5 — Manual authentication mechanism 5

7.5 Mechanism 6 – Devices with simple input capabilities

7.5.1 Specific requirements

This mechanism has the following specific requirements.

a) The mechanism specified in this subclause is appropriate for the case where both devices (A and B) have a simple input interface.

b) One of the devices (the device labelled A below) shall have the means to generate (long) random keys and short random bit-streams.

7.5.2 Specification of data exchanged

The following data exchanges and operations shall take place (see also Figure 6). Note that steps a)–b) may occur in parallel, as may steps d)–e).

a) Devices A and B shall provisionally agree on a data string D. This could, for example, be achieved via an exchange of unprotected messages sent via the shared communications link.

b) Device A shall generate and keep secret a (long) random key k and a (short) random bit-stream R. Device A shall compute $h(D||k||R)$ and then pass this hash-value to device B by some (not necessarily secure) means, e.g. via the shared communications link.

c) Both devices shall output a signal via their output interfaces to acknowledge that they have completed steps a)–b), and that they are ready for the authentication mechanism to commence. On observing the signals, the user shall enter a signal into device A via its simple input interface to notify A that the mechanism can start.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

d) Device A shall transmit the (long) key k and (short) random bit-stream R to device B via the shared communications link.

e) Device A shall output the short random bit-stream R on its standard output interface.

f) On receipt of k and R in step c), device B shall re-compute the hash-value $h(D||k||R)$ as a function of its stored version of data D. If this hash-value equals the value device B received from A in step b) then device *B* shall output the short random bit-stream R via its standard output interface. Otherwise B shall give a failure signal, and shall implement a mechanism whereby it will refuse to start a new instance of the mechanism for a short time-period.

g) The user shall compare the two short random bit-streams output via the standard output interfaces of devices A and B in steps e)–f). If they are the same then the user shall enter a signal of acceptance into both devices. Otherwise the mechanism has failed and the user enters rejection signal into the devices via their simple input interfaces. The devices interpret the absence of an acceptance signal as a failure signal, this will require the implementation of a time-out mechanism.

NOTE The time delay mechanism used in steps f) prevents a man-in-the-middle attack in which an attacker tries to masquerade as device *A* to device *B* immediately after *B* abandons the run while *A* is still waiting for an acceptance signal in step g).

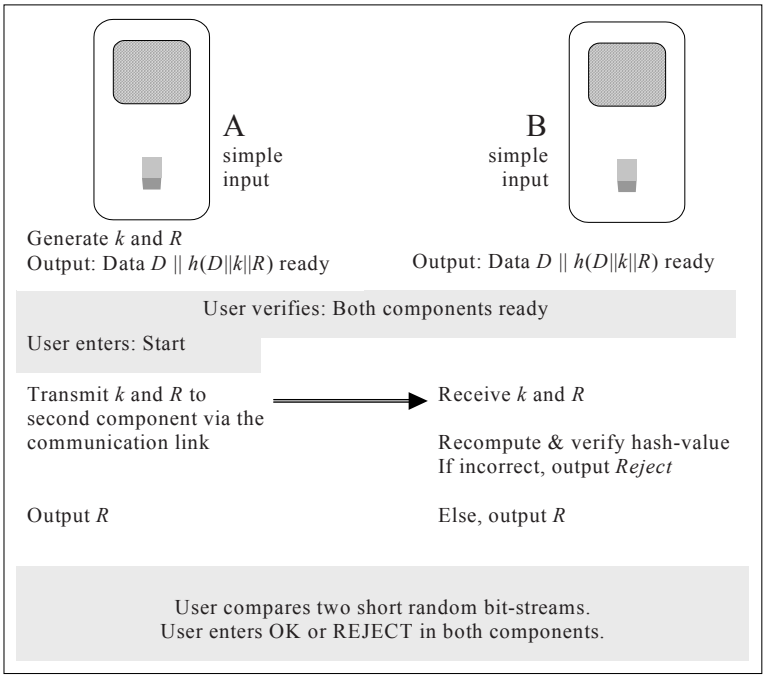

Figure 6 — Manual authentication mechanism 6

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

8 Mechanisms using a MAC

8.1 General

In this clause two manual authentication mechanisms are specified that are based on the use of a Message Authentication Code (MAC). The two mechanisms are appropriate for different types of devices. Specifically,

— the first mechanism (mechanism 7) is appropriate for the case where both devices have a simple output interface, and

— the second mechanism (mechanism 8) is appropriate for the case where one device has a simple input interface and the other has a simple output interface

A standard input or output interface can emulate a simple interface, and hence if both devices have standard input and output interfaces then either of the mechanisms may be used.

Both mechanisms operate in the following general way. A data string D is transferred from one device to the other (or is the concatenation of data transferred in both directions) via the shared communications link. The manual entity authentication mechanism is then executed. As a result of the mechanism both devices are provided with assurance that the data string D they possess is the same as the value held by the other device.

8.2 Mechanism 7 – Devices with simple output capabilities

8.2.1 General

This mechanism has two variants (7a and 7b). Mechanism 7a, specified in clause 8.2.3, requires fewer interactions between the two devices, whereas mechanism 7b, specified in clause 8.2.4, requires less manual user interaction.

8.2.2 Specific requirements

This mechanism has the following specific requirements.

a) The two variants of the mechanism specified in this subclause are appropriate for the case where both devices (A and B) have a simple output interface.

b) Both devices shall have the means to generate random MAC keys, and the user shall have the means to generate short random bit-strings.

c) The devices know each other's identity before starting the mechanism.

NOTE If the user makes poor choices for the random bit-string, e.g. the user always chooses the same value, then there is a greatly increased risk of a successful attack on the mechanism.

8.2.3 Specification of data exchanged in mechanism 7a

The following data exchanges and operations shall take place (see also Figure 7). Note that steps b) and c) may occur in parallel, as may steps d)–e) and f)–g).

a) Both devices shall output, via their respective simple output interfaces, a signal to acknowledge that they have received data D and that they are ready for the authentication mechanism to commence. On observing that both devices are ready, the user shall generate a short random bit-string R. The user shall enter the random bit-string R into both devices, and shall then enter a signal into device A to notify A that the mechanism can start.

b) Device A shall generate a random key K_A, where K_A is suitable for use as a key with the MAC function shared by the two devices. Using K_A as the key, device A computes a MAC (labelled MAC_A) on the data string made up of the concatenation of I_A (an identifier for A), the data D, and the random bit-string R. Device A shall transmit MAC_A to device B via the shared communications link.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

c) Device B shall generate a random key K_B, where K_B is suitable for use as a key with the MAC function shared by the two devices. Using K_B as the key, device B computes a MAC (labelled MAC_B) on the data string made up of the concatenation of I_B (an identifier for B), the data D, and the bit string R. Device B shall transmit MAC_B to device A via the shared communications link.

d) Once device A has received MAC_B (and not before), device A shall send K_A to device B.

e) On receipt of K_A, device B verifies that MAC_A equals a MAC value computed using the stored values of R, D, I_A, and the received value K_A as the key. If verification is successful, device B outputs an indication of success.

f) Once device B has received MAC_A (and not before), device B shall send K_B to device A.

g) On receipt of K_B, device A verifies that MAC_B equals a MAC value computed using the stored values of R, D, I_B and the received value K_B as the key. If verification is successful, device A outputs an indication of success.

h) The user verifies that both devices have given an indication of success, and, if so, enters a confirmation of success into both devices. If one or both of the devices give a failure indication, then the user shall enter a failure indication into both devices. If the user fails to enter a success notification into a device within a specified time interval, then this shall be interpreted as a failure of the mechanism.

NOTE Step g) in this mechanism prevents a substitution attack where the attacker tries to masquerade as device A to device B.

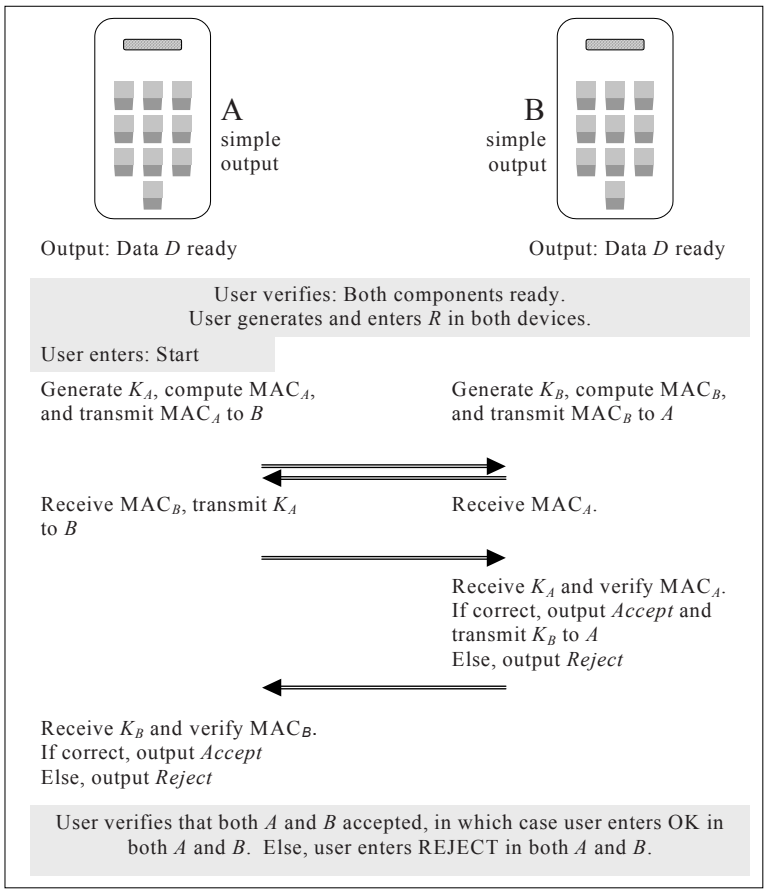

Figure 7 — Manual authentication mechanism 7a

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

8.2.4 Specification of data exchanged in mechanism 7b

The following data exchanges and operations shall take place (see also Figure 8).

a) Both devices shall output a signal to acknowledge that they have received data D and that they are ready for the authentication mechanism to commence. On observing that both devices are ready, the user shall generate a short random bit-string $R = (r_1,r_2,\ldots,r_n)$, where r_i is a bit and n is the number of bits in R. The user shall enter the value R into both devices, and shall then enter a signal into device A to notify A that the mechanism can start.

b) For i set consecutively to 1, 2, …, n, steps 1)–5) shall be executed. (Note that steps 1) and 2) may be executed in parallel).

   1) Device A shall generate a random key K_{Ai}, where K_{Ai} is suitable for use as a key with the MAC function shared by the two devices. Using K_{Ai} as the key, device A computes a MAC (labelled MAC_{Ai}) on the data string made up of the concatenation of I_A (an identifier for A), the data D, and the random bit r_i. Device A shall transmit MAC_{Ai} to device B via the shared communications link.

   2) Device B shall generate a random key K_{Bi}, where K_{Bi} is suitable for use as a key with the MAC function shared by the two devices. Using K_{Bi} as the key, device B computes a MAC (labelled MAC_{Bi}) on the data string made up of the concatenation of I_B (an identifier for B), the data D, and the random bit r_i. Device B shall transmit MAC_{Bi} to device A via the shared communications link.

   3) On receipt of MAC_{Bi}, device A sends K_{Ai} to device B.

   4) On receipt of MAC_{Ai} and K_{Ai}, device B verifies that MAC_{Ai} equals a MAC value computed using the stored values of r_i, D, I_A, and the received key K_{Ai}. If verification is successful, device B sends K_{Bi} to device A; otherwise it aborts the protocol.

   5) On receipt of K_{Bi}, device A verifies that MAC_{Bi}, as received in step 3, equals a MAC value computed using the stored values of r_i, D, I_B and the received key K_{Bi}. If verification is unsuccessful, device A aborts the protocol.

NOTE In the case $i = n$, if verification is successful in steps 4) and 5), devices *B* and *A*, respectively, can output an indication of success. Whilst this is not an integral part of the protocol, it may be useful to give the device user an indication that the process has completed successfully.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

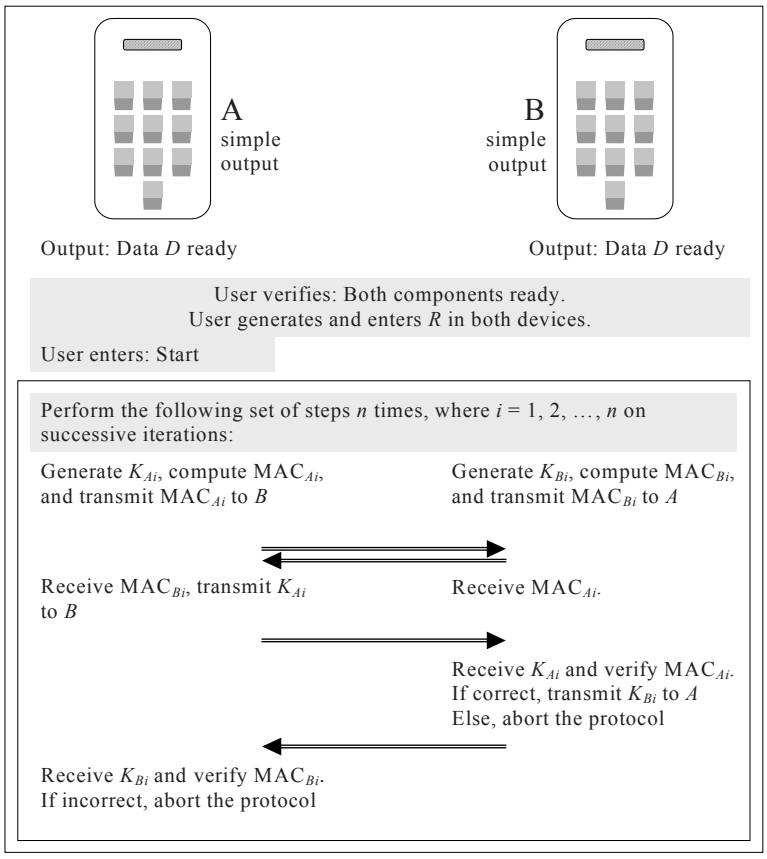

Figure 8 — Manual authentication mechanism 7b

8.3 Mechanism 8 – One device with simple input, one device with simple output

8.3.1 General

This mechanism has two variants (8a and 8b). Mechanism 8a requires fewer interactions between the two devices, whereas mechanism 8b requires less manual user interaction.

8.3.2 Specific requirements

This mechanism has the following specific requirements.

a) The two variants of the mechanism specified in this subclause are appropriate for the case where one device (A) has a simple input interface and the other device (B) has a simple output interface.

b) Both devices shall have the means to generate random MAC keys.

c) The devices know each other's identity before starting the mechanism.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

8.3.3 Specification of data exchanged in mechanism 8a

The data exchanges and operations are precisely the same as those for Mechanism 7a (as specified in clause 8.2.3) with the following exception:

— In step a) device A generates the random bit-string and displays it to the user, who copies it into device B. Thus in this mechanism the user is not required to generate a random bit-string.

This mechanism is shown in Figure 9.

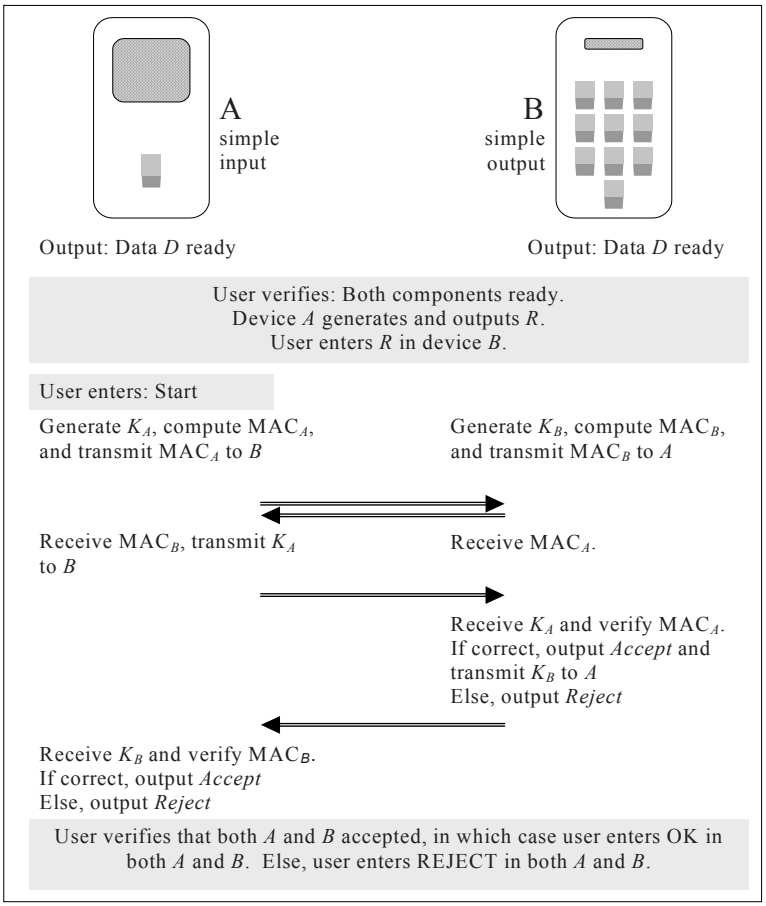

Figure 9 — Manual authentication mechanism 8a

8.3.4 Specification of data exchanged in mechanism 8b

The data exchanges and operations are precisely the same as those for mechanism 7b (as specified in clause 8.2.4) with the following exception:

— In step a) device A generates the random bit-string and displays it to the user, who copies it into device B. Thus in this mechanism the user is not required to generate a random bit-string.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Annex A
(normative)

ASN.1 modules

A.1 Formal definition

```
EntityAuthenticationMechanisms-6 {

   iso(1) standard(0) e-auth-mechanisms(9798) part6(6)

      asn1-module(0) object-identifiers(0) }

   DEFINITIONS EXPLICIT TAGS ::= BEGIN

-- EXPORTS All; --
-- IMPORTS None; --

OID ::= OBJECT IDENTIFIER -- alias

-- Synonyms --

is9798-6 OID ::= { iso(1) standard(0) e-auth-mechanisms(9798) part6(6) }

mechanism OID ::= { is9798-6 mechanisms(1) }

-- Mechanisms using manual transfer of a short key and a short check-value --

mdt-kc-siso OID ::= {mechanism mdt-kc-siso(1)}

mdt-kc-sisi OID ::= {mechanism mdt-kc-sisi(2)}

-- Mechanisms using manual transfer of a short digest-value or a short key --

mdt-c-siso-one OID ::= {mechanism mdt-c-sisoone(3)}

mdt-c-siso-two OID ::= {mechanism mdt-c-sisotwo(4)}

mdt-c-sisi-one OID ::= {mechanism mdt-c-sisione(5)}

mdt-c-sisi-two OID ::= {mechanism mdt-c-sisitwo(6)}

-- Mechanisms using a MAC --

mac-k-soso OID ::= {mechanism mac-k-soso(7)}

mac-k-siso OID ::= {mechanism mac-k-siso(8)}

END -- EntityAuthenticationMechanisms-6 --
```


© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Annex B
(informative)

Using manual authentication protocols for the exchange of secret keys

B.1 General

In this annex we describe methods for enabling devices to share a secret key using one of the manual authentication mechanisms specified in the body of this part of ISO/IEC 9798.

B.2 Authenticated Diffie-Hellman key agreement

The procedure described here is the Diffie-Hellman key agreement mechanism, which is authenticated using a manual authentication mechanism. The key agreement mechanism conforms to key agreement mechanism 4 of ISO/IEC 11770-3 (see also Annex B.5 of ISO/IEC 11770-3). The description given below is simplified, and for a full description the reader is referred to ISO/IEC 11770-3.

The Diffie-Hellman key agreement mechanism is described in terms of a general group G (expressed in multiplicative terminology), and an element g in G, which has a sufficiently large order. The steps of the procedure are as follows.

a) Device A generates randomly and privately an integer x, computes g^x and sends it to device *B*.

b) Device B generates randomly and privately an integer y, computes g^y and sends it to device A.

c) Devices A and B execute one of the manual authentication protocols for data D = (g^x || g^y || text) where 'text' is any additional data, e.g., each others' identifiers, that the devices may want to agree upon.

d) If the result of the manual authentication protocol is successful, then the components can compute the shared Diffie-Hellman key as $S = g^{xy}$.

The components can then derive secret cryptographic keys of the required length and format from the shared secret Diffie-Hellman key S.

B.3 Authenticated Diffie-Hellman key agreement using a manual authentication certificate

B.3.1 General

The procedure described here is the Diffie-Hellman key agreement mechanism, where one of the Diffie-Hellman public keys is authenticated using a manual authentication certificate (hence this procedure is specific to mechanism 1 and the requirements specified in subclause 6.2.1 shall be met). Device B is authenticated to device A using an encrypted version of the check-value key K used in the mechanism. Note that this mechanism requires the two devices to agree and implement a symmetric encryption mechanism *e*, where $e_L(M)$ denotes the encryption of data M using secret key L. Symmetric encryption techniques are standardized in ISO/IEC 18033-3 and ISO/IEC 18033-4.

The Diffie-Hellman key agreement mechanism is described in terms of a general group G (expressed in multiplicative terminology), and an element g in G, which has a sufficiently large order. The procedure has two stages, Stage 1 and Stage 2.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

In Stage 1, device A generates its Diffie-Hellman private key, computes the corresponding public key, and produces a manual authentication certificate for a data string including this public key. The certificate is transferred to device B. In Stage 2, device B receives the public key of device A, verifies it, and generates its own private and public Diffie-Hellman keys. Further, both devices compute the shared Diffie-Hellman secret, from which a secret encryption key is derived. Finally, device A verifies the encrypted version of the key K it receives from device B as part of the manual authentication process. As the result, the Diffie-Hellman secret shared by the two devices has been authenticated.

B.3.2 Stage 1

a) Device A generates randomly and privately an integer x and computes g^x. Device A then creates a manual authentication certificate on the data string D made up of g^x and any other data that needs to be reliably transferred to device B. The manual authentication certificate (K, check-value) is manually transferred to device B, and device B stores it. Device A stores x and g^x, and any other data items included within D.

B.3.3 Stage 2 (initiated by either device at some later time)

b) Device A sends g^x and possibly some other data to device B via the shared communications link. Device B verifies the authenticity of g^x based on the stored manual authentication certificate.

c) Device B generates randomly and privately an integer y, and computes g^y. Device B computes the Diffie-Hellman shared secret as $S = (g^x)^y$ and uses S to encrypt the key K, i.e. the key from the manual authentication certificate. Device B sends the encrypted key $e_S(K)$ and its Diffie-Hellman public key g^y to device A.

d) Device A computes its copy of the shared secret as $S = (g^y)^x$. Then it decrypts $e_S(K)$ and verifies that K is correct. If so, then device A can accept S as authenticated.

The two devices can then derive cryptographic keys of the required length and format from the shared secret Diffie-Hellman key S.

NOTE In the above description the key K from the manual authentication certificate was used in steps c) and d). Any other appropriate value, e.g., the check-value or some special purpose value, agreed between the parties at Stage 1, could be used instead.

B.4 More than two components

Means by which more than two devices can agree on a secret key using manual authentication techniques are now described.

a) One device acts as a ‘master’ device.

b) The master device executes the mechanism described in B.2 with every other device to establish a shared secret key with each of them.

c) Then any of the devices can generate the common secret key, which is then distributed encrypted and possibly also integrity-protected to the remainder of the devices via the master device.

If the number of devices is n, the master device needs to compute n exponentiations in the group G. Each of the other devices needs to compute two exponentiations. Therefore it is advisable to choose the master device to have sufficient computing power to perform the task.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Annex C
(informative)

Using manual authentication protocols for the exchange of public keys

C.1 General

In this annex methods are described that enable devices to reliably exchange a public key using one of the manual authentication mechanisms specified in the body of this part of ISO/IEC 9798. The context of the description is between a Certification Authority (CA) and a client of the CA. The CA needs to reliably transfer its public key to the client, and the client needs to reliably transfer its public key to the CA.

Two different cases are described, depending on whether the CA client generates its own private keys, or whether they are generated by a key management facility and then imported to the device.

C.2 Requirements

The CA shall be equipped with a standard output interface, e.g. a display, and a simple input interface for giving it commands. The CA client shall possess a standard input interface and a simple output interface, e.g. an audio output, to indicate success or failure of the process.

NOTE It is straightforward to adapt the procedure to the case where the CA and CA client have different types of user interfaces. Only the type of the manual authentication protocol needs to be changed.

The CA client and the CA share a communications channel.

C.3 Private key generated in device

The procedure operates as follows.

a) The CA shall be reliably informed of the identifier for the CA client. This could, for example, be achieved by the user entering the identifier for the client into the input interface of the CA. However, it could also be achieved as part of the protocol itself (see below).

b) The CA sends its public key P_{CA} to the CA client, and the CA client sends its public key P_M to the CA. This transfer is assumed to take place via the (untrusted) communications link. Along with P_M, the CA client can send any other information it wishes to have included in the public key certificate, which will be generated by the CA. This could, for example, include the identifier for the client.

c) The CA and the client now perform the manual authentication mechanism specified in subclause 6.2 to verify that the exchanged public keys are correct. The CA takes the role of device A, and the client the role of device B. The data D used within the manual authentication mechanism consists of P_{CA}, P_M, and any other data supplied by the client and CA. This additional data may include the unique identifiers of the CA and the client.

d) If (and only if) the client (device B) emits a success indication, the user instructs the CA (device A) to generate an appropriate public key certificate. This certificate can then be sent (possibly unprotected) to the client via the communications link.

e) The client (device B) now performs two checks before accepting the certificate. Firstly, the client verifies the signature using the CA's public key (P_{CA}). Secondly the client verifies that the data fields within the certificate (including the public key P_M and the client identifier) are all as expected. The procedure is now complete.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

C.4 Private key generated externally

If the private key of the CA client is generated by the CA or any other trusted key generation service, then the private key shall be securely transported to the client.

A procedure for transporting a private key from the CA to the client is now described. The steps of the procedure are as follows:

a) The CA and the client establish a shared secret key as described in Annex B.

b) Using the secret key established in step a), the CA sends the client private key encrypted and integrity protected to the client, where it is stored securely. The CA also sends its public key (P_{CA}) integrity protected to the client, again using the secret key established in step a). Symmetric encryption techniques are standardized in ISO/IEC 18033-3 and ISO/IEC 18033-4.

c) The client now sends any information it wants to have included in its certificate to the CA, integrity protected using the secret key established in step a).

d) The CA generates the certificate for the client's public key P_M. This certificate can then be sent (possibly unprotected) to the client via the communications link.

e) The client now performs two checks before accepting the certificate. Firstly the client checks the signature using the CA's public key (P_{CA}). Secondly the client verifies that the data fields within the certificate (including the public key P_M and the identifier for the client) are all as expected. The procedure is now complete.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Annex D
(informative)

On mechanism security and choices for parameter lengths

D.1 General

In this annex the security of the eight manual authentication mechanisms specified in this standard is discussed. Guidance is also provided on choices for the lengths of check-values, digest-values, MACs, random bit-strings, and keys.

D.2 Use of mechanisms 1 and 2

All data to be transferred via the communications link between the two devices is assumed to be public, even if in some cases part of data D may be secret. The security goal of the manual authentication mechanisms is to protect the integrity of the data, not its confidentiality. The necessary integrity protection is performed using a checking procedure based on the check-value.

A check-value function is a mapping, f, from a data space, D, and a key space, K, to a check-value space, C:

$$f : D \times K \to C, \qquad c = f(d,k) .$$

In mechanisms 1 and 2 check-values are used to protect the integrity of data. Therefore the security of these mechanisms is based on the unconditional security of the check-value function rather than computational security. The unconditional security of check-value functions is based on results developed by *message authentication theory,* see, for example, Section 4.5 of [28]. Two main types of attack are normally considered:

- Impersonation attacks, and
- Substitution attacks.

In an *impersonation attack*, the attacker tries to convince a receiver that data was sent by a legitimate sender without observing any prior data exchange between the sender and the receiver. In a *substitution attack*, the attacker first observes some data d and then *replaces* it with some other data $\hat{d} \neq d$. The probabilities for the attacker to succeed in an impersonation attack and a substitution attack are denoted by P_I and P_S, respectively, and they can be expressed as

$$P_I \hat{=} \max_{c \in C, d \in D} P_k(c = f(d,k)),$$

$$P_S \hat{=} \max_{\substack{c,\hat{c} \in C \\ d,\hat{d} \in D, d \neq \hat{d}}} P_k(\hat{c} = f(\hat{d},k) \mid c = f(d,k)) .$$

The security of both Mechanisms depends on the probability of an attacker successfully replacing the observed data d with some other data $\hat{d} \neq d$. The attacker succeeds if $\hat{d}$ is accepted by the component as valid data. Since we assume that the two devices are physically close to each other and we do not accept any data unless both devices signal that they are ready, the impersonation attack does not apply to the manual authentication scenario. Furthermore, the normal situation for integrity protection using a MAC is that *both* the data *and* the MAC are transmitted and can be observed by the attacker. This is not the case for Mechanisms 1 and 2, which is why a check-value is used instead of a MAC in these mechanisms. Here only the data is sent over a public channel, and the attacker does not know the output of the check-value function until after

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

the data D has been transferred (in fact, in mechanism 1 the attacker will never have access to the output of this function). This simplifies the security analysis and the expression for a successful substitution attack. Hence, the probability of successful substitution attack for Mechanisms 1 and 2 can be expressed as

$$P_S = \max_{\substack{d,\hat{d}\in D \\ d\neq\hat{d}}} P(f(d,k) = f(\hat{d},k) \mid d \text{ is observed})$$

Thus, given that the key k is chosen uniformly at random from the key space, K, the probability above can be expressed as

$$P_S = \max_{d,\hat{d}\in D, d\neq\hat{d}} \frac{|\{k \in K : f(d,k) = f(\hat{d},k)\}|}{|K|},$$

where $|K|$ denotes the cardinality of the set K. It follows from this equation that in order to provide high security, the collision probability of the check-value function shall be low. This can be guaranteed by using check-value functions obtained from error correcting codes, such as the scheme specified in Annex E.

Based on the above analysis, a key length of 16–20 bits and a check-value length of 16–20 bits are recommended. Tables giving the probabilities of successful attacks for 16 and 20 bit lengths are provided in Annex E.

D.3 Use of mechanisms 3 and 5

The security of mechanisms 3 and 5 is based on similar principles to those on which mechanisms 1 and 2 are based. Instead of a check-value function, a digest function and a hash-function shall be employed with these mechanisms. Hash-functions are standardized in ISO/IEC 10118 [11], and a hash-function from this standard is recommended for use with Mechanisms 3 and 5.

Since the long random key k is used to ensure that the second condition (i.e. digest collision) of a digest function is satisfied, a key k of 160 bits is recommended for this case.

Since digest-values are manually transferred or compared between devices A and B, a short digest function whose output is of b = 16–20 bits is recommended for this case. A digest function is similar to a MAC algorithm except that a digest-value is shorter than the typical length of a MAC. As a result, a possible digest construction involves use of the first b bits of either a MAC or a standard cryptographic hash-function output, as suggested in [25] and in Annex G.

Proofs of security for mechanisms 3 and 5 are given in [24].

D.4 Use of mechanisms 4 and 6

The security of mechanisms 4 and 6 is based on different principles to those on which mechanisms 1 and 2 are based. Instead of a check-value function, a hash-function shall be employed with these mechanisms. Hash-functions are standardized in ISO/IEC 10118 [11], and a hash-function from this standard is recommended for use with mechanisms 4 and 6.

As for mechanisms 3 and 5, a key length for k of 160 bits is recommended for this case. These mechanisms involve the manual transfer or comparison of a short bit-stream R between devices A and B, and therefore a bit length for R of 16–20 bits is recommended for this case.

Proofs of security for mechanisms 4 and 6 are given in [24].

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

D.5 Use of mechanisms 7 and 8

The security of mechanisms 7 and 8 is based on different principles to those on which mechanisms 1 and 2 are based. Instead of a check-value function, a MAC function shall be employed with these mechanisms. MAC functions are standardized in ISO/IEC 9797, and a MAC function from this standard is recommended for use with Mechanisms 7 and 8.

A random bit-string R of 16–20 bits is recommended for this case, but the MAC should be longer. The size of the output of the MAC function to be used for mechanisms 7 and 8 should be in the range 128–160 bits. Similarly, the random keys K_A and K_B (and K_{Ai} and K_{Bi}), used as keys for the MAC function, should be of approximately the same size, i.e. 128–160 bits. Time-out procedures should also be implemented to detect possible interruptions to the mechanism.

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

© ISO/IEC 2010 – All rights reserved

Annex E
(informative)

A method for generating short check-values

E.1 General

In this annex a check-value function suitable for use with mechanisms 1 and 2 is specified. The probabilities of successful attacks on these mechanisms when the proposed check-value scheme is employed are also considered. Using the expression for successful attack in Annex D, a straightforward approach is to use check-value functions derived from coding theory. The relationship between error correcting codes and such check-values is discussed in [15].

Before considering concrete examples, two basic definitions from coding theory are given. For simplicity, only codes defined over a finite field $\boldsymbol{F_q}$ are considered. Denote a q-ary code over $\boldsymbol{F_q}$ by V. Suppose the codewords have length n. The code is a mapping from messages to codewords. Each message has its corresponding unique codeword. Then the code V consists of all vectors $\boldsymbol{v} \in V = \{\boldsymbol{v}^{(d)} : d \in D\}$, $\boldsymbol{v}^{(d)} = (v_1(d), v_2(d), ..., v_n(d))$, where $v_i(d) \in \boldsymbol{F_q}$.

Two further definitions are required.

Definition: If x and y are two q-ary tuples of length n, then we say that their *Hamming-distance* is

$$d_H(x,y) \hat{=} |\{i \in \{1,...,n\} : x_i \neq y_i\}|.$$

Definition: The *minimum distance* of a code V is

$$d_H(V) \hat{=} \min_{x,y \in V, x \neq y} d_H(x,y).$$

We now show how to create a check-value function suitable for use with Mechanisms 1 and 2 based on a code. The construction is very simple, and the mapping from the message and key space is simply obtained as

$$f(d,k) = v_k(d),$$

where $k \in K = \{1,... ,n\}$. Hence, a check-value function is obtained with a key size equal to n and with message space size equal to the coding space size.

The probability of a successful substitution attack for this construction is determined as follows. From the expression for P_S in Annex D, it immediately follows that:

$$P_S = \max_{d,\hat{d} \in D, d \neq \hat{d}} \frac{|\{k \in K : f(d,k) = f(\hat{d},k)\}|}{|K|} = \max_{d,\hat{d} \in D, d \neq \hat{d}} \frac{|\{k \in K : v_k(d) = v_k(\hat{d})\}|}{|K|}$$

$$= \max_{d,\hat{d} \in D, d \neq \hat{d}} \frac{n - d_H(v^{(d)}, v^{(\hat{d})})}{n} = 1 - \frac{d_H(V)}{n}.$$

Given this exact expression for the probability of successful substitution attack, it is now appropriate to consider some concrete constructions. Rather long codes with very high minimum distance are required. This property holds for the well-known Reed-Solomon (RS) codes [25]. An RS code can be constructed over an arbitrary finite field, $\boldsymbol{F_q}$. The calculation of a codeword is very simple and involves polynomial evaluation over

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

the finite field. Express the data (message) to be encoded as a q-tuple of length t over $\boldsymbol{F}_q$, $d = d_0, d_1, .., d_{t-1}, d_i \in \boldsymbol{F}_q$. Then, the generalized RS encoding polynomial is given by

$$p^{(d)}(x) = d_0 + d_1 x + d_2 x^2 + \ldots + d_{t-1} x^{t-1}.$$

The check-value function is directly given by evaluating the polynomial at an arbitrary point $k \in \boldsymbol{F}_q$,

$$f(d,k) = v_k(d) = p^{(d)}(k) = d_0 + d_1 k + d_2 k^2 + \ldots + d_{t-1} k^{t-1}.$$

The generalized RS code has the following properties ([25]):

$n = q = |K|$,

$|D| = q^t = n^t$,

$d_H(V) = n - t + 1$.

This implies that $P_S = (t\text{-}1)/n$ for a check-value obtained from the generalized RS code. The probability increases with the size of the message space, D. Hence, a good approach is to **first** apply a good one-way hash-function, such as one of the dedicated hash-functions specified in ISO/IEC 10118-3, to the data and then use the output from the one-way hash-function as input to the Reed-Solomon code. This implies that we keep a low probability without significantly increasing the key length or the length of the output of the check-value function. By using this approach, a message space of around 128 bits (truncated SHA-1) gives sufficient security. In Table 1 two construction examples and the corresponding probabilities of successful attacks are given.

Table E.1 — RS code check-values: probability of successful substitution attack, P_S

$\log_2\|D\|$	$\log_2(n)$	P_S
128	16	2^{-13}–2^{-16}
256	16	2^{-12}–2^{-16}
128	20	2^{-17}–2^{-20}
256	20	2^{-16}–2^{-20}

As can be seen from the table, a code with a 4 hexadecimal digit key and check-value gives a forgery probability of around 2^{-12} or less. If this is increased to 5 hexadecimal digits, the probability decreases to around 2^{-17} or less.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Annex F
(informative)

Comparative analysis in security and efficiency of mechanisms 1–8

Since mechanisms 1, 2, 7 and 8 were standardized in 2005 (mechanisms 7 and 8 were labelled as 3 and 4), a number of better schemes have been proposed. These include schemes which use less user input and achieve tighter bounds on the probability of success for an attacker. Moreover, some mechanisms are now available which possess proofs of security, whereas mechanisms 1, 2, 7 and 8 do not possess such strong evidence of security (although neither are we aware of any problems with these mechanisms).

We note in particular the work of Laur and Nyberg [19], Vaudenay and Pasini [25] and [30], Cagalj, Capkun and Hubaux [1], Wong and Stajano [32] and [33], Hoepman [6] and [7], Lindell [20], and Nguyen and Roscoe [21], [23] and [24]. Although a number of different schemes have been proposed, they all fall into one of two categories with respect to the way that the authenticated data D is processed by the cryptographic functions used as was pointed out in [24]. Both approaches are represented in the new mechanisms included in this second edition, i.e. (1) mechanisms 3 and 5 use a (short output) digest function to bind D to a long random key k; and (2) mechanisms 4 and 6 use a (long output or cryptographic) hash-function to bind D to a long random key k and a short random bit-stream R.

In order that users of this standard have access to the best available techniques, four new mechanisms (mechanisms 3–6) have been included in clause 7. These mechanisms offer more efficient and better alternatives to mechanisms 1, 2, 7 and 8 while they do not require any changes in the input and output interfaces of the devices conducting the protocols.

All the newly included mechanisms 3–6 have been published, as have their proofs of security – see [21], [22], [23], [24], [32] and [33].

The main features of the four newly included mechanisms are as follows.

— Mechanisms 3–6 halve the amount of manual data transfer or comparison required in mechanisms 1–2. In mechanisms 1 and 2, the user has to manually transfer both a short key and a check-value from one device to the other, or compare by eye those values displayed by the devices. Halving the amount of human interactions, in mechanisms 3–6, human users only need to transfer or compare either a short digest-value or a short random bit-stream both of which are of equal length as a short key or a check-value in mechanisms 1–2.

  NOTE 1 Manual transfers of a short random key, a check-value or a digest-value are more significant than simpler actions such as those of pressing buttons to start the protocols, reading 1-bit results (Accept/Reject), or any other sing bit comparisons which occur in all mechanisms. The latter type of human interactions is therefore ignored in this analysis and in Table F.1.

— Mechanisms 3–6 offer a higher level of security than mechanisms 1–2 even though they only require half the human effort required in mechanisms. It was demonstrated in [21] and [24] that mechanisms 1–2 do not offer as much security as would ideally be the case because of the short bit length of the key K which is instrumental in the computation of the check-value. In contrast, the key k used in mechanisms 3–6 can be sent over the (high-bandwidth) shared communications link. Hence, it can be significantly longer, i.e. 160 bits as specified in Annex D. This property derives from theoretical lower bounds for the key-length of universal hash-functions, as have been studied extensively in, for example, [2], [5], [27], and [31].

— Mechanisms 3–6 possess proofs of security given in [21] and [24]. Although mechanisms 3–6 are different from one another, they provide the same level of security relative to an equal amount of human effort.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

— Although mechanisms 7–8 provide the same level of security as mechanisms 3–6, they rely on secret manual data transfers of a short random key R. Unlike mechanisms 1–6, the random key in mechanisms 7–8 shall be kept secret, so that is known only by the devices and the human user. Care, therefore, shall be taken during manual data transfers to avoid the random key being observed, e.g. via a hidden camera. If the secret key is compromised, an intruder may be able to launch a man-in-the-middle attack.

Table F.1 summarises the difference in manual data transfer and security between mechanisms 1–8. Here, a successful attack implies a protocol run where devices compute the same bit string which is manually transferred, however the adversary has successfully manipulated data D when it was exchanged over the shared (insecure) communications link so that devices received different versions of data D.

NOTE 2 In mechanisms 1, 2, 3, 5, 7 and 8, keys used with a check-value function, a digest function or a MAC algorithm are always random and fresh in each protocol session. Substitution attacks, which rely on the reuse of a single key for multiple message inputs as mentioned in Annex D.2, are therefore irrelevant. More information about proofs of security and the definitions of a successful attack can be found in [21] and [24].

In order to compare the amount of manual data transfer, the bit lengths of a digest-value (mechanisms 3 and 5), a short random bit-stream R (mechanisms 4, 6, 7 and 8), a check-value and a short secret key used with a check-value function (mechanisms 1 and 2) are all of b bits.

Since there are two types of human interactions involved in the mechanisms: (1) manual transfers of bits of information (such as digest-values and short random bit-streams); and (2) manual comparisons of two bit-streams, Table F.1 will use the notation (m, c) to represent the number of bits of manual data transfer (m) and the number of bits of comparison (c).

Table F.1 — Comparison between mechanisms 1–8

Mechanism	Types of devices A and B	Human interactions (in public/secret bits)	Cryptographic functions	Probability of a successful attack
1 (old)	A: simple input interface B: simple output interface	$(2b, 0)$ (public data)	check-value function	$> 2^{-b}$
2 (old)	A,B: simple input interface	$(0, 2b)$ (public data)	check-value function	$> 2^{-b}$
3 (new)	A: simple input interface B: simple output interface	$(b, 0)$ (public data)	hash-function & digest function	$2^{-b} + \varepsilon$
4 (new)	A: simple input interface B: simple output interface	$(b, 0)$ (public data)	hash-function	$2^{-b} + \varepsilon$
5 (new)	A,B: simple input interface	$(0, b)$ (public data)	hash-function & digest function	$2^{-b} + \varepsilon$
6 (new)	A,B: simple input interface	$(0, b)$ (public data)	hash-function	$2^{-b} + \varepsilon$
7 (old)	A,B: simple output interface	$(2b, 0)$ (secret data)	MAC algorithm	$2^{-b} + \varepsilon$
8 (old)	A: simple input interface B: simple output interface	$(b, 0)$ (secret data)	MAC algorithm	$2^{-b} + \varepsilon$

NOTE 3 Mechanisms 1, 2, 7 and 8 were standardised in 2005 and so are marked "(old)" in Table F.1. In contrast, the newly added mechanisms 3–6 are marked "(new)".

NOTE 4 Manual transfers of public data or bits imply that the transferred data can be known to any one as in mechanisms 1–6. This is the opposite of mechanisms 7–8 which keep the manually transferred short key secret to any one except the devices and the human user.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

NOTE 5 We classify all mechanisms relative to their required cryptographic functions because the information will become useful when we compare mechanisms 3–6 in terms of their computational complexity in Annex F.2.

NOTE 6 In Table F.1, ε is a value that is negligible relative to 2^{-b}, whereas "$> 2^{-b}$" indicates that the probability of a successful attack is greater than 2^{-b} by a non-negligible value.

Among the newly added mechanisms, there is also a potential advantage in computational efficiency of mechanisms 3 and 5 over mechanisms 4 and 6. This can be explained by the following observation which is taken from [16], [21], [22] and [23]. Data D, which nodes want to authenticate, only needs to be processed by a short output digest function (e.g. b = 16–20) in mechanisms 3 and 5 as opposed to a long output hash-function (e.g. 160 or more bits) in mechanisms 4 and 6. Since data D is normally large in practice, e.g. it may consist of images or DVDs which are significantly longer than a random key *k*, it is potentially faster to compute a short digest of *D* than a long hash-value. Hence, mechanisms 3 and 5 can be more efficient than mechanisms 4 and 6 in terms of computational cost. This is particularly useful if these mechanisms are implemented on small devices and lightweight cryptographic application, where optimising computational efficiency is an important criterion.

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Annex G
(informative)

Methods for generating short digest-values

In this annex two constructions for digest functions suitable for use with mechanisms 3 and 5 are specified. The first is derived from a MAC algorithm, such as those standardized in ISO/IEC 9797; the second is derived from a hash-function, such as those standardized in ISO/IEC 10118.

NOTE 1 There are a variety of other mechanisms which can also be used to compute digest functions, such as one based on Toeplitz matrix multiplication or integer multiplication as suggested in references [17], [23] and [21]. However, these constructions for digest functions have not been fully analysed and tested, and hence they are not described in this Annex.

In the definitions below, the $trunc_b(x)$ function gives as output the leading (leftmost) b bits of the bit-string x.

Definition 1: to compute a digest-value of message m using key k, compute

$$d(m, k) = trunc_b(\ \mathrm{MAC}_k(m)\ )$$

Definition 2: to compute a digest-value of message m using key k, compute

$$d(m, k) = trunc_b(\ h(m||k)\ )$$

NOTE 2 The method given in definition 1 is proposed in [4], and the method given in definition 2 is specified in [22].

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Bibliography

[1] M. Cagalj, S. Capkun, and J. Hubaux, 'Key agreement in peer-to-peer wireless networks', in: *Proceedings of the IEEE, Special Issue on Security and Cryptography* **94**(2) (2006), 467–478

[2] J.L. Carter and M.N. Wegman, 'Universal Classes of Hash Functions', *Journal of Computer and System Sciences* **18**(2) (1979), 143–154

[3] C. Gehrmann and K. Nyberg, 'Enhancements to Bluetooth baseband security', in: *Proceedings of Nordsec* 2001, Copenhagen, Denmark, November 2001

[4] C. Gehrmann, C.J. Mitchell and K. Nyberg, 'Manual authentication for wireless devices', *Cryptobytes* **7**(1) (2004), 29–37

[5] P. Gemmell and M. Naor, 'Codes for Interactive Authentication', in: *Advances in Cryptology – Crypto 1993*, LNCS, Vol. 773, D.R. Stinson, ed., Springer, 1993, pp. 355–367

[6] J.-H. Hoepman, 'Ephemeral Pairing on Anonymous Networks', in *Proceedings of the Second International Conference on Security in Pervasive Computing (SPC 2005)*, LNCS, Vol. 3450, D. Hutter and M. Ullmann, eds., Springer, 2005, pp. 101–116

[7] J.-H. Hoepman, 'Ephemeral Pairing Problem', in: *Proceeding of the 8th International Conference on Financial Cryptography*, LNCS, Vol. 3110, A. Juels, ed., Springer, 2004, pp. 212–226

[8] ISO 7498-2:1989, *Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture*

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

[10] ISO/IEC 8825-1, *Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*

[11] ISO/IEC 10118 (all parts), *Information technology — Security techniques — Hash-functions*

[12] ISO/IEC 11770-3:2008, *Information technology — Security techniques — Key management — Part 3: Mechanisms using asymmetric techniques*

[13] ISO/IEC 18033-3:2005, *Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers*

[14] ISO/IEC 18033-4:2005, *Information technology — Security techniques — Encryption algorithms — Part 4: Stream ciphers*

[15] G. Kabatianskii, B. Smeets and T. Johansson, 'On the cardinality of systematic authentication codes via error correcting codes', *IEEE Transactions on Information Theory* **42**(2) (1996), 566–578

[16] K. Khoo, F.-L. Wong and C.-W. Lim, 'On a construction of short digests for authenticating ad-hoc networks', in: *Proceedings of ICCSA 2009*, LNCS vol. 5593, pp. 863-876

[17] H. Krawczyk, 'New Hash Functions For Message Authentication', in: *Advances in Cryptology – Eurocrypt 1995*, LNCS, Vol. 921, L.C. Guillou and J.-J. Quisquater, eds., Springer, 1995, pp. 301–310

[18] J.-O. Larsson, 'Higher layer key exchange techniques for Bluetooth security', in: *Opengroup Conference*, Amsterdam, October 2001

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

[19] S. Laur and K. Nyberg, 'Efficient Mutual Data Authentication Using Manually Authenticated Strings', LNCS, Vol. 4301, D. Pointcheval, ed., Springer, 2006, pp. 90–107

[20] A.Y. Lindell. 'Comparison-Based Key Exchange and the Security of the Numeric Comparison Mode in Bluetooth v2.1', in: *Proceedings of the Cryptographers' Track at the RSA Conference 2009 on Topics in Cryptology*, LNCS, Vol. 5473, M. Fischlin, ed., Springer, 2009, pp. 66–83

[21] L. H. Nguyen and A. W. Roscoe, 'Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey', *Journal of Computer Security* (to appear). See: http://eprint.iacr.org/2010/206.pdf

[22] L. H. Nguyen and A. W. Roscoe, 'Authenticating ad hoc networks by comparison of short digests', *Information & Computation* **206**(2–4) (2008), 250–271

[23] L. H. Nguyen and A. W. Roscoe, 'Efficient group authentication protocol based on human interaction' in: *Proceedings of Workshop on Foundation of Computer Security and Automated Reasoning Protocol Security Analysis*, 2006, pp. 9–31. See: http://eprint.iacr.org/2009/150

[24] L. H. Nguyen and A. W. Roscoe, 'Separating two roles of hashing in one-way message authentication', in *Proceedings of Workshop on Foundation of Computer Security, Automated Reasoning Protocol Security Analysis, and Issues in the Theory of Security*, 2008, pp. 195–209. See: http://eprint.iacr.org/2009/003

[25] S. Pasini and S. Vaudenay, 'SAS-based Authenticated Key Agreement', in *Proceedings of International Conference on Practice and Theory in Public Key Cryptography (PKC 2006)*, LNCS, Vol. 3958, M. Yung, Y. Dodis, A. Kiayias and T. Malkin, eds., Springer, 2006, pp. 395–409

[26] I.S. Reed and G. Solomon, 'Polynomial codes over certain finite fields'*, SIAM Journal* **8** (1960), 300–304

[27] D.R. Stinson, 'Universal Hashing and Authentication Codes', in *Advances in Cryptology – Crypto 1991*, LNCS, Vol. 576, J. Feigenbaum, ed., Springer, 1992, pp. 74–85

[28] D.R. Stinson, 'Cryptography – Theory and Practice', *CRC Press*, 2002, 2nd edition

[29] SHAMAN Project Deliverable D13 (Annex 2), *Final technical report – Workpackage 2 – Security for distributed terminals*, 2003. Available at http://www.ist-shaman.org/

[30] S. Vaudenay, 'Secure Communications over Insecure Channels Based on Short Authenticated Strings', in: *Advances in Cryptology – Crypto 2005*, LNCS, Vol. 3621, V. Shoup, ed., Springer, 2005, pp. 309-326

[31] M.N. Wegman and J.L. Carter, 'New Hash Functions and Their Use in Authentication and Set Equality', *Journal of Computer and System Sciences* **22**(3) (1981), 265–279

[32] F.-L. Wong and F. Stajano, 'Multi-channel Protocols', in *Proceedings of the 13th International Workshop on Security Protocols*, LNCS, Vol. 4631, B. Christianson, B. Crispo, J.A. Malcolm and M. Roe, eds., Springer, 2005, pp. 128–132

[33] F.-L. Wong and F. Stajano, 'Multi-channel Security Protocols'*, IEEE Pervasive computing* **6**(4) (2007), 31–39

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

ICS 35.040
Price based on 35 pages

© ISO/IEC 2010 – All rights reserved

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.

This page deliberately left blank

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

PLUS is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

BSI Group Headquarters

389 Chiswick High Road London W4 4AL UK

Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

Copyright

All the data, software and documentation set out in all British Standards and other BSI publications are the property of and copyrighted by BSI, or some person or entity that owns copyright in the information used (such as the international standardization bodies) and has formally licensed such information to BSI for commercial publication and use. Except as permitted under the Copyright, Designs and Patents Act 1988 no extract may be reproduced, stored in a retrieval system or transmitted in any form or by any means – electronic, photocopying, recording or otherwise – without prior written permission from BSI. Details and advice can be obtained from the Copyright & Licensing Department.

Useful Contacts:

Customer Services
Tel: +44 845 086 9001
Email (orders): orders@bsigroup.com
Email (enquiries): cservices@bsigroup.com

Subscriptions
Tel: +44 845 086 9001
Email: subscriptions@bsigroup.com

Knowledge Centre
Tel: +44 20 8996 7004
Email: knowledgecentre@bsigroup.com

Copyright & Licensing
Tel: +44 20 8996 7070
Email: copyright@bsigroup.com

bsi.

...making excellence a habit.™

Licenced to Care Segreteria - BOLOGNA , CRUI Conferenza dei Rettori delle downloaded on 23/04/2025. For status of this document please check https://knowledge.bsigroup.com. Reproduction and distribution is prohibited.