INTERNATIONAL STANDARD

ISO/IEC 9798-2

Second edition
1999-07-15

Information technology — Security techniques — Entity authentication —

Part 2:
Mechanisms using symmetric encipherment algorithms

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

Partie 2: Mécanismes utilisant des algorithmes de chiffrement symétriques

Reference number
ISO/IEC 9798-2:1999(E)

Contents

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. 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.

International Standard ISO/IEC 9798-2 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 9798-2:1994), which has been technically revised. Note, however, that implementations which comply with ISO/IEC 9798-2 (1st edition) will be compliant with ISO/IEC 9798-2 (2nd 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*

Further parts may follow.

Annex A of this part of ISO/IEC 9798 is for information only.

Information technology — Security techniques — Entity authentication —

Part 2:
Mechanisms using symmetric encipherment algorithms

1 Scope

This part of ISO/IEC 9798 specifies entity authentication mechanisms using symmetric encipherment algorithms. Four of the mechanisms provide entity authentication between two entities where no trusted third party is involved; two of these are mechanisms to unilaterally authenticate one entity to another, while the other two are mechanisms for mutual authentication of two entities. The remaining mechanisms require a trusted third party for the establishment of a common secret key, and realize mutual or unilateral entity authentication.

The mechanisms specified in this part of ISO/IEC 9798 use time variant parameters such as time stamps, sequence numbers, or random numbers, to prevent valid authentication information from being accepted at a later time or more than once.

If no trusted third party is involved and a time stamp or sequence number is used, one pass is needed for unilateral authentication, while two passes are needed to achieve mutual authentication. If no trusted third party is involved and a challenge and response method employing random numbers is used, two passes are needed for unilateral authentication, while three passes are required to achieve mutual authentication. If a trusted third party is involved, any additional communication between an entity and the trusted third party requires two extra passes in the communication exchange.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9798. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/IEC 9798 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/IEC 9798-1:1997, *Information technology — Security techniques — Entity authentication — Part 1: General*.

ISO/IEC 11770-2:1996, *Information technology — Security techniques — Key management — Part 2: Mechanisms using symmetric techniques.*

3 Definitions and notation

For the purposes of this part of ISO/IEC 9798, the definitions and notation described in ISO/IEC 9798-1 apply.

4 Requirements

In the authentication mechanisms specified in this part of ISO/IEC 9798 an entity to be authenticated corroborates its identity by demonstrating its knowledge of a secret authentication key. This is achieved by the entity using its secret key to encipher specific data. The enciphered data can be deciphered by anyone sharing the entity's secret authentication key.

The authentication mechanisms have the following requirements. If any one of these is not met then the authentication process may be compromised or it cannot be implemented.

a) A claimant authenticating itself to a verifier shall share a common secret authentication key with that verifier, in which case the mechanisms of clause 5 apply, or each entity shall share a secret authentication key with a common trusted third party, in which case the mechanisms of clause 6 apply. Such keys shall be known to the involved parties prior to the commencement of any particular run of an authentication mechanism. The method by which this is achieved is beyond the scope of this part of ISO/IEC 9798.

b) If a trusted third party is involved it shall be trusted by both the claimant and the verifier.

c) The secret authentication key shared by a claimant and a verifier, or by an entity and a trusted third party, shall be known only to those two parties and, possibly, to other entities they both trust.

NOTE 1 The encipherment algorithm and the key lifetime should be chosen so that it is computationally infeasible for a key to be deduced during its lifetime. In addition, the key lifetime should be chosen to prevent known plaintext or chosen plaintext attacks.

a) For every possibility for the secret key *K*, the encipherment function *eK* and its corresponding decipherment function *dK* shall have the following property. The decipherment process *dK*, when applied to a string *eK*(*X*), shall enable the recipient of that string to detect forged or manipulated data, i.e. only the possessor of the secret key *K* shall be capable of generating strings which will be 'accepted' when subjected to the decipherment process *dK*.

NOTE 2 In practice, this can be achieved in many ways. Two examples are as follows.

1. If sufficient redundancy is present in, or appended to, the data, and the encipherment algorithm is chosen with care, the integrity requirement can be satisfied. The redundancy is checked for correctness by the recipient before the deciphered data can be accepted as valid.

2. The key *K* is used to derive a pair of keys *K'* and *K''*. The key *K''* is then used to calculate a Message Authentication Code (MAC) on the data to be enciphered, while the key *K'* is used to encipher the data concatenated with the MAC. The recipient checks that the value of the MAC is correct before accepting the deciphered data as valid.

a) The mechanisms in this part of ISO/IEC 9798 require the use of time variant parameters such as time stamps, sequence numbers or random numbers. The properties of these parameters, in particular that it is most unlikely for them to repeat within the lifetime of a secret authentication key, are important for the security of these mechanisms. For additional information see annex B of ISO/IEC 9798-1.

5 Mechanisms not involving a trusted third party

In these authentication mechanisms the entities *A* and *B* shall share a common secret authentication key K_{AB} or two unidirectional secret keys K_{AB} and K_{BA} prior to the commencement of any particular run of the authentication mechanisms. In the latter case the unidirectional keys K_{AB} and K_{BA} are used respectively for the authentication of *A* by *B* and of *B* by *A*.

All text fields specified in the following mechanisms are available for use in applications outside the scope of this part of ISO/IEC 9798 (they may be empty). Their relationship and contents depend upon the specific application. See annex A for information on the use of text fields.

5.1 Unilateral authentication

Unilateral authentication means that only one of the two entities is authenticated by use of the mechanism.

5.1.1 One pass authentication

In this authentication mechanism the claimant *A* initiates the process and is authenticated by the verifier *B*. Uniqueness/timeliness is controlled by generating and checking a time stamp or a sequence number (see annex B of ISO/IEC 9798-1). The authentication mechanism is illustrated in Figure 1.

Figure 1

The form of the token (Token*AB*), sent by the claimant *A* to the verifier *B* is:

$$\text{Token}AB = \text{Text2} \,\|\, eK_{AB}\left(\begin{matrix} T_A \\ N_A \end{matrix} \,\|\, B \,\|\, \text{Text1}\right)$$

where the claimant *A* uses either a sequence number N_A or a time stamp T_A as the time variant parameter. The choice depends on the technical capabilities of the claimant and the verifier as well as on the environment.

The inclusion of the distinguishing identifier *B* in Token*AB* is optional.

> NOTE Distinguishing identifier *B* is included in Token*AB* to prevent the re-use of Token*AB* on entity *A* by an adversary masquerading as entity *B*. Its inclusion is made optional so that, in environments where such attacks cannot occur, it may be omitted.
>
> The distinguishing identifier *B* may also be omitted if a unidirectional key is used.

(1) *A* generates and sends Token*AB* to *B*.

(2) On receipt of the message containing Token*AB*, *B* verifies Token*AB* by deciphering the enciphered part (where deciphering implies that the requirements of 4.d are met) and then checking the correctness of the distinguishing identifier *B*, if present, as well as the time stamp or the sequence number.

5.1.2 Two pass authentication

In this authentication mechanism the claimant *A* is authenticated by the verifier *B* who initiates the process. Uniqueness/timeliness is controlled by generating and checking a random number R_B (see annex B of ISO/IEC 9798-1). The authentication mechanism is illustrated in Figure 2.

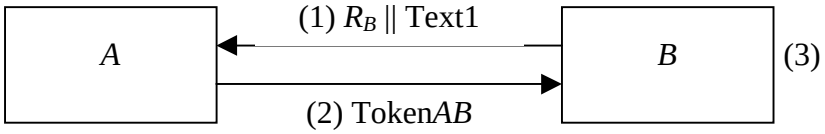

Figure 2

The form of the token (Token*AB*), sent by the claimant *A* to the verifier *B* is:

$$\text{Token}AB = \text{Text3} \,\|\, eK_{AB}(R_B \,\|\, B \,\|\, \text{Text2}) \,.$$

The inclusion of the distinguishing identifier *B* in Token*AB* is optional.

> NOTE 1 In order to prevent the possibility of a known plaintext attack, i.e. a cryptanalytic attack where the cryptanalyst knows the complete plaintext for one or more ciphertext strings, entity *A* may include a random number R_A in Text2.

NOTE 2 Distinguishing identifier *B* is included in Token*AB* to prevent a so-called reflection attack. Such an attack is characterised by the fact that an intruder 'reflects' the challenge R_B to *B* pretending to be *A*. The inclusion of the distinguishing identifier *B* is made optional so that, in environments where such attacks cannot occur, it may be omitted.

The distinguishing identifier *B* may also be omitted if a unidirectional key is used.

(1) *B* generates a random number R_B and sends it and, optionally, a text field Text1 to *A*.

(2) *A* generates and sends Token*AB* to *B*.

(3) On receipt of the message containing Token*AB*, *B* verifies Token*AB* by deciphering the enciphered part (where deciphering implies that the requirements of 4.d are met) and then checking the correctness of the distinguishing identifier *B*, if present, and that the random number R_B, sent to *A* in step (1), agrees with the random number contained in Token*AB*.

5.2 Mutual authentication

Mutual authentication means that the two communicating entities are authenticated to each other by use of the mechanism.

The two mechanisms described in 5.1.1 and 5.1.2 are adapted in 5.2.1 and 5.2.2, respectively, to achieve mutual authentication. In both cases this requires one more pass and results in two more steps.

NOTE A third mechanism for mutual authentication can be constructed from two instances of the mechanism specified in 5.1.2, one started by entity *A* and the other by entity *B*.

5.2.1 Two pass authentication

In this authentication mechanism uniqueness/timeliness is controlled by generating and checking time stamps or sequence numbers (see annex B of ISO/IEC 9798-1).

The authentication mechanism is illustrated in Figure 3.

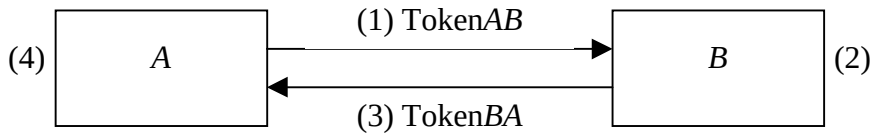

Figure 3

The form of the token (Token*AB*), sent by *A* to *B*, is identical to that specified in 5.1.1.

$$\text{Token}AB = \text{Text2} \,\|\, eK_{AB}\left(\begin{matrix} T_A \\ N_A \end{matrix} \,\|\, B \,\|\, \text{Text1}\right).$$

The form of the token (Token*BA*), sent by *B* to *A*, is:

$$\text{Token}BA = \text{Text4} \,\|\, eK_{AB}\left(\begin{matrix} T_B \\ N_B \end{matrix} \,\|\, A \,\|\, \text{Text3}\right).$$

The inclusion of the distinguishing identifier *B* in Token*AB* and the inclusion of the distinguishing identifier *A* in Token*BA* are (independently) optional.

NOTE 1 Distinguishing identifier *B* is included in Token*AB* to prevent the re-use of Token*AB* on entity *A* by an adversary masquerading as entity *B*. For similar reasons the distinguishing identifier *A* is present in Token*BA*. Their inclusion is made optional so that, in environments where such attacks cannot occur, one or both may be omitted.

The distinguishing identifiers *A* and *B* may also be omitted if unidirectional keys (see below) are used.

The choice of using either time stamps or sequence numbers in this mechanism depends on the capabilities of the claimant and the verifier as well as on the environment.

Steps (1) and (2) are identical to those specified in 5.1.1, one pass authentication.

(3) *B* generates and sends Token*BA* to *A*.

(4) The message in step (3) is handled in a manner analogous to step (2) of 5.1.1.

NOTE 2 The two messages of this mechanism are not bound together in any way, other than implicitly by timeliness; the mechanism involves independent use of mechanism 5.1.1 twice. Further binding together of these messages can be achieved by making appropriate use of text fields.

If unidirectional keys are used then the key K_{AB} in Token*BA* is replaced by the unidirectional key K_{BA}, and the appropriate key is used in step (4).

5.2.2 Three pass authentication

In this authentication mechanism uniqueness/timeliness is controlled by generating and checking random numbers (see annex B of ISO/IEC 9798-1).

The authentication mechanism is illustrated in Figure 4.

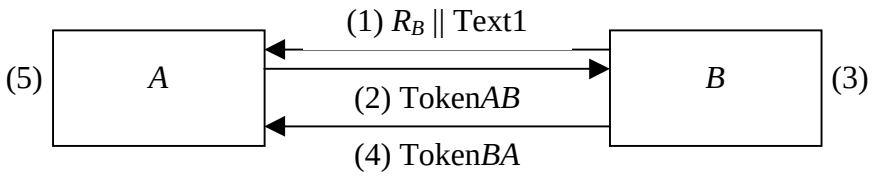

Figure 4

The tokens are of the following form:

$$\text{Token}AB = \text{Text3} \,||\, eK_{AB}(R_A \,||\, R_B \,||\, B \,||\, \text{Text2}),$$

$$\text{Token}BA = \text{Text5} \,||\, eK_{AB}(R_B \,||\, R_A \,||\, \text{Text4}).$$

The inclusion of the distinguishing identifier *B* in Token*AB* is optional.

NOTE When present, distinguishing identifier *B* is included in Token*AB* to prevent a so-called reflection attack. Such an attack is characterized by the fact that an intruder 'reflects' the challenge R_B to *B* pretending to be *A*. The inclusion of the distinguishing identifier *B* is made optional so that, in environments where such attacks cannot occur, it may be omitted.

The distinguishing identifier *B* may also be omitted if unidirectional keys (see below) are used.

(1) *B* generates a random number R_B and sends it and, optionally, a text field Text1 to *A*.

(2) *A* generates a random number R_A, and generates and sends Token*AB* to *B*.

(3) On receipt of the message containing Token*AB*, *B* verifies Token*AB* by deciphering the enciphered part (where deciphering implies that the requirements of 4.d are met) and then checking the correctness of the distinguishing identifier *B*, if present, and that the random number R_B, sent to *A* in step (1), agrees with the random number contained in Token*AB*.

(4) *B* generates and sends Token*BA* to *A*.

(5) On receipt of the message containing Token*BA*, *A* verifies Token*BA* by deciphering the enciphered part (where deciphering implies that the requirements of 4.d are met) and then checking that the random number R_B, received from *B* in step (1) agrees with the random number contained in Token*BA* and that the random number R_A, sent to *B* in step (2), agrees with the random number contained in Token*BA*.

If unidirectional keys are used then the key K_{AB} in Token*BA* is replaced by the unidirectional key K_{BA}, and the appropriate key is used in step (5).

6 Mechanisms involving a trusted third party

The authentication mechanisms in this clause do not make use of a secret key shared by the two entities prior to the authentication process. They do, however, make use of a trusted third party (with distinguishing identifier *TP*) with which the entities *A* and *B* each share a secret key, K_{AT} and K_{BT} respectively. In each mechanism one of the entities requests a key K_{AB} from the trusted third party. This is followed by an adaptation of the mechanisms described in 5.2.1 and 5.2.2, respectively.

As described below certain passes may be omitted from each mechanism if only unilateral authentication is required.

All text fields specified in the following mechanisms are available for use in applications outside the scope of this part of ISO/IEC 9798 (they may be empty). Their relationship and contents depend upon the specific application. See annex A for information on the use of text fields.

6.1 Four pass authentication

In this mutual authentication mechanism uniqueness/timeliness is controlled by using time variant parameters (see annex B of ISO/IEC 9798-1). This mechanism is equivalent to Key Establishment Mechanism 8 of ISO/IEC 11770-2: 1996.

The authentication mechanism is illustrated in Figure 5.

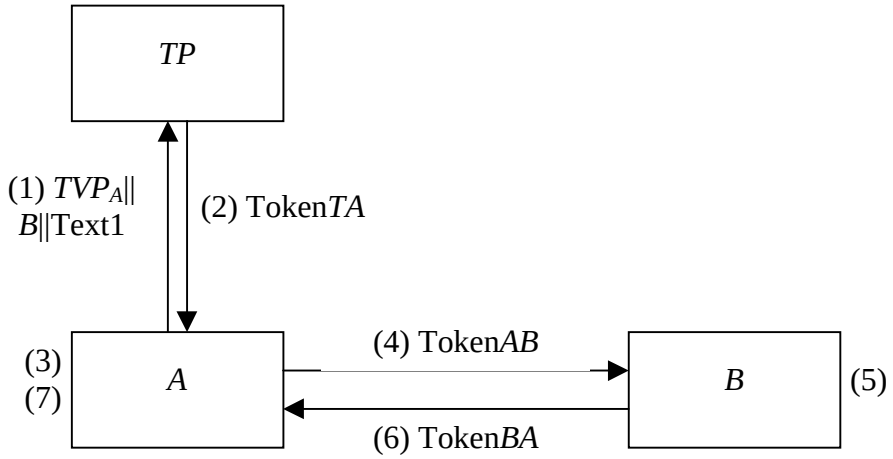

Figure 5

The form of the token (Token*TA*), sent by *TP* to *A*, is:

$$\text{Token}TA = \text{Text4} \,\|\, eK_{AT}(TVP_A \,\|\, K_{AB} \,\|\, B \,\|\, \text{Text3}) \,\|\, eK_{BT}\left(\begin{matrix} T_{TP} \\ N_{TP} \end{matrix} \,\|\, K_{AB} \,\|\, A \,\|\, \text{Text2}\right).$$

The form of the token (Token*AB*), sent by *A* to *B*, is:

$$\text{Token}AB = \text{Text6} \,\|\, eK_{BT}\left(\begin{matrix} T_{TP} \\ N_{TP} \end{matrix} \,\|\, K_{AB} \,\|\, A \,\|\, \text{Text2}\right) \|\, eK_{AB}\left(\begin{matrix} T_A \\ N_A \end{matrix} \,\|\, B \,\|\, \text{Text5}\right).$$

The form of the token (Token*BA*), sent by *B* to *A*, is:

$$\text{Token}BA = \text{Text8} \,\|\, eK_{AB}\left(\begin{matrix} T_B \\ N_B \end{matrix} \,\|\, A \,\|\, \text{Text7}\right).$$

The choice of using either time stamps or sequence numbers in this mechanism depends on the capabilities of the entities involved as well as on the environment.

The use of the time variant parameter TVP_A in steps (1) through (3) of figure 5, as specified below, is somewhat different from its normal use. It allows *A* to associate the response message (2) with the message request (1). The important property of the time variant parameter here is its non-repeatability, to limit the possible re-use of a previously used Token*TA*.

NOTE The time variant parameter TVP_A could be a random number. However, unlike the random numbers used in certain of the mechanisms in this part of ISO/IEC 9798, it is not necessary that TVP_A be unpredictable to a third party, and a non-repeating counter value would be equally appropriate.

(1) *A* generates a time variant parameter TVP_A, and sends it, the distinguishing identifier *B* and, optionally, a text field Text1 to the trusted third party *TP*.

(2) The trusted third party *TP* generates and sends Token*TA* to *A*.

(3) On receipt of the message containing Token*TA*, *A* verifies Token*TA* by deciphering the data enciphered under K_{AT} (where deciphering implies that the requirements of 4.d are met) and then checking the correctness of the distinguishing identifier *B* and that the time variant parameter, sent to *TP* in step (1), agrees with the time variant parameter contained in Token*TA*. In addition, *A* retrieves the secret authentication key K_{AB}. *A* then extracts

$$eK_{BT}\left(\begin{matrix} T_{TP} \\ N_{TP} \end{matrix} \,\|\, K_{AB} \,\|\, A \,\|\, \text{Text2}\right)$$

from Token*TA* and uses it to construct Token*AB*.

(4) *A* generates and sends Token*AB* to *B*.

(5) On receipt of the message containing Token*AB*, *B* verifies Token*AB* by deciphering the enciphered parts (where deciphering implies that the requirements of 4.d are met) and then checking the correctness of the distinguishing identifiers *A* and *B* as well as the time stamp(s) or the sequence number(s). In addition, *B* retrieves the secret authentication key K_{AB}.

(6) *B* generates and sends Token*BA* to *A*.

(7) On receipt of the message containing Token*BA*, *A* verifies Token*BA* by deciphering the enciphered part (where deciphering implies that the requirements of 4.d are met) and then checking the correctness of the distinguishing identifier *A* as well as the time stamp or the sequence number.

Steps (6) and (7) may be omitted if only unilateral authentication of *A* to *B* is required.

6.2 Five pass authentication

In this mutual authentication mechanism uniqueness/timeliness is controlled by using random numbers (see annex B of ISO/IEC 9798-1). This mechanism is equivalent to Key Establishment Mechanism 9 of ISO/IEC 11770-2: 1996.

The authentication mechanism is illustrated in Figure 6.

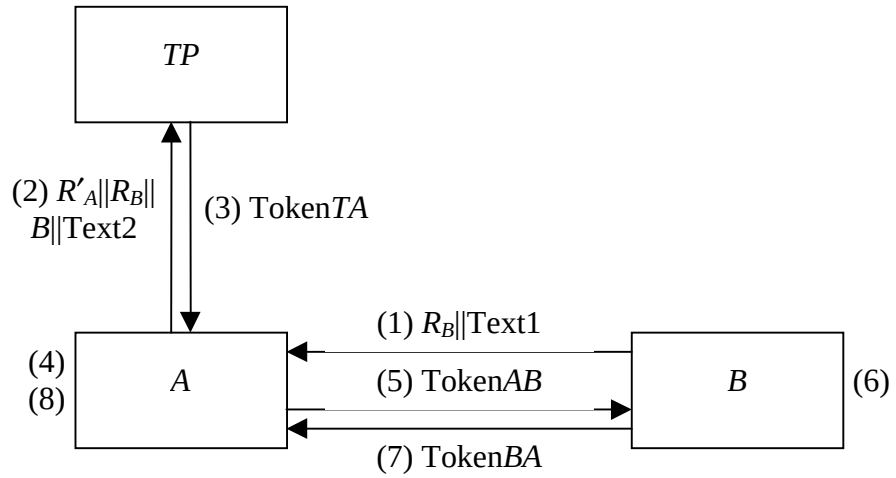

Figure 6

The form of the token (Token*TA*), sent by *TP* to *A*, is:

$$\text{Token}TA = \text{Text5} \,||\, eK_{AT}(R'_A \,||\, K_{AB} \,||\, B \,||\, \text{Text4}) \,||\, eK_{BT}(R_B \,||\, K_{AB} \,||\, A \,||\, \text{Text3}).$$

The form of the token (Token*AB*), sent by *A* to *B*, is:

$$\text{Token}AB = \text{Text7} \,||\, eK_{BT}(R_B \,||\, K_{AB} \,||\, A \,||\, \text{Text3}) \,||\, eK_{AB}(R_A \,||\, R_B \,||\, \text{Text6}).$$

The form of the token (Token*BA*), sent by *B* to *A*, is:

$$\text{Token}BA = \text{Text9} \,||\, eK_{AB}(R_B \,||\, R_A \,||\, \text{Text8}).$$

(1) *B* generates a random number R_B and sends it and, optionally, a text field Text1 to *A*.

(2) *A* generates a random number R'_A and sends it, the random number R_B, the distinguishing identifier *B* and, optionally, a text field Text2 to the trusted third party *TP*.

(3) The trusted third party *TP* generates and sends Token*TA* to *A*.

(4) On receipt of the message containing Token*TA*, *A* verifies Token*TA* by deciphering the data enciphered under K_{AT} (where deciphering implies that the requirements of 4.d are met) and then checking the correctness of the distinguishing identifier *B* and that the random number R'_A, sent to *TP* in step (2), agrees with the random number contained in Token*TA*. In addition, *A* retrieves the secret authentication key K_{AB}. *A* then extracts

$$eK_{BT}(R_B \,||\, K_{AB} \,||\, A \,||\, \text{Text3})$$

from Token*TA* and uses it to construct Token*AB*.

(5) *A* generates a second random number R_A, and generates and sends Token*AB* to *B*.

(6) On receipt of the message containing Token*AB*, *B* verifies Token*AB* by deciphering the enciphered parts (where deciphering implies that the requirements of 4.d are met) and then checking the correctness of the distinguishing identifier *A* and that the random number R_B, sent to *A* in step (1), agrees with both copies contained in Token*AB*. In addition, *B* retrieves the secret authentication key K_{AB}.

(7) *B* generates and sends Token*BA* to *A*.

(8) On receipt of the message containing Token*BA*, *A* verifies Token*BA* by deciphering the enciphered part (where deciphering implies that the requirements of 4.d are met) and then checking that the random number R_B, received from *B* in step (1), agrees with the random number contained in Token*BA* and that the random number R_A, sent to *B* in step (5), agrees with the random number contained in Token*BA*.

Steps (7) and (8) may be omitted if only unilateral authentication of *A* to *B* is required.

Annex A
(informative)

Use of text fields

The tokens specified in clauses 5 and 6 of this part of ISO/IEC 9798 contain text fields. The actual use of and the relationships between the various text fields in a given pass depend on the application. Some examples are given below; see also annex A of ISO/IEC 9798-1.

If the tokens do not contain (sufficient) redundancy, the enciphered text fields may be used to provide additional redundancy.

Any information requiring confidentiality or data origin authentication should be placed in the enciphered part of the token.

Bibliography

[1] ISO/IEC 9797: 1994, *Information technology — Security techniques — Data integrity mechanism using a cryptographic check function employing a block cipher algorithm*.

[2] ISO/IEC 10118-1: 1994, *Information technology — Security techniques — Hash-functions — Part 1: General*.

[3] ISO/IEC 10118-2: 1994, *Information technology — Security techniques — Hash-functions — Part 2: Hash-functions using an n-bit block cipher algorithm*.

ICS 35.040

Price based on 11 pages