INTERNATIONAL STANDARD

ISO/IEC 13888-2

First edition
1998-04-01

Information technology — Security techniques — Non-repudiation —

Part 2:
Mechanisms using symmetric techniques

Technologies de l'information — Techniques de sécurité — Non-répudiation —

Partie 2: Mécanismes utilisant des techniques symétriques

Reference number
ISO/IEC 13888-2:1998(E)

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 13888-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

ISO/IEC 13888 consists of the following parts, under the general title *Information technology — Security techniques — Non-repudiation*:

— Part 1: General

— Part 2: Mechanisms using symmetric techniques

— Part 3: Mechanisms using asymmetric techniques

Further parts may follow.

Annex A of this part of ISO/IEC 13888 is for information only.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

Information technology — Security techniques — Non-repudiation —

Part 2:

Mechanisms using symmetric techniques

1 Scope

The goal of the non-repudiation service is to generate, collect, maintain, make available and validate evidence concerning a claimed event or action in order to resolve disputes about the occurrence or non occurrence of the event or action. This part of ISO/IEC 13888 provides descriptions of generic structures that can be used for non-repudiation services, and of some specific, communication related mechanisms which can be used to provide non-repudiation of origin (*NRO*), non-repudiation of delivery (*NRD*), non-repudiation of submission (*NRS*), and non-repudiation of transport (*NRT*) services. Other non-repudiation services can be built using the generic structures described in Clause 8 in order to meet the requirements defined by the security policy.

This part of ISO/IEC 13888 relies on the existence of a trusted third party (*TTP*) to prevent fraudulent repudiation. Usually an on-line trusted third party is needed.

Non-repudiation mechanisms provide protocols for the exchange of non-repudiation tokens specific to each non-repudiation service. Non-repudiation tokens used in this part consist of Secure Envelopes and additional data. Non-repudiation tokens shall be stored as non-repudiation information that may be used subsequently in case of disputes.

Depending on the non-repudiation policy in effect for a specific application, and the legal environment within which the application operates, additional information may be required to complete the non-repudiation information, e.g.,

- evidence including a trusted time stamp provided by a Time Stamping Authority,
- evidence provided by a notary which provides assurance about the action or event performed by one or more entities.

Non-repudiation can only be provided within the context of a clearly defined security policy for a particular application and its legal environment. Non-repudiation policies are described in ISO/IEC 10181-4.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 13888. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 13888 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

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

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

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

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

ISO/IEC 10181-4:1997, *Information technology — Open Systems Interconnection — Security frameworks for open systems — Part 4: Non-repudiation framework.*

ISO/IEC 13888-1:1997, *Information technology — Security techniques — Non-repudiation — Part 1: General.*

3 Definitions

For the purposes of this part of ISO/IEC 13888, the definitions described in ISO/IEC 13888-1 apply.

4 Notation and Abbreviations

4.1 Notation

4.1.1 Notation from ISO/IEC 13888-1

Imp(y) *imprint of data string y, either (1) the hash-code of data string y, or (2) the data string y.*

$SENV_X$ *the secure envelope generated with the secret key x of entity X.*

text — *a data item forming part of a token that may contain additional information, e.g., key identifier and/or the message identifier.*

y||z — *the result of the concatenation of y and z in that order.*

4.1.2 Notation unique for the purposes of this part of ISO/IEC 13888

a — *a secret key known only to entity* A *and a TTP.*

A — *the distinguishing identifier of entity* A.

b — *a secret key known only to entity B and a TTP.*

B — *the distinguishing identifier of entity* B.

da — *secret key of the Delivery Authority DA.*

f, f_i — *data item (flag) indicating the kind of non-repudiation service in effect.*

m — *a message for which evidence is generated.*

$MAC_X(y)$ — *the cryptographic check value computed on the data y using the key of entity X.*

T_g — *date and time the evidence was generated.*

T_i — *date and time the event or action took place.*

ttp — *a secret key known only to the TTP to generate non-repudiation tokens.*

x — *a secret key known either to two entities or only to the trusted third party.*

z_1 — *a data field consisting of data fields relevant for the provision of the NRO token.*

z_2 — *a data field consisting of data fields relevant for the provision of the NRD token.*

z_3 — *a data field consisting of data fields relevant for the provision of the NRS token.*

z_4 — *a data field consisting of data fields relevant for the provision of the NRT token.*

z_5 — *a data field consisting of data fields relevant for the provision of the TST.*

4.2 Abbreviations

DA	*Delivery Authority.*
GNRT	*Generic Non-Repudiation Token.*
NRD	*Non-Repudiation of Delivery.*
NRDT	*Non-repudiation of delivery token.*
NRO	*Non-Repudiation of Origin.*
NROT	*Non-repudiation of origin token.*
NRS	*Non-Repudiation of Submission.*
NRST	*Non-repudiation of submission token.*
NRT	*Non-Repudiation of Transport.*
NRTT	*Non-repudiation of transport token.*
Pol	*the distinguishing identifier of the non-repudiation policy (or policies) which apply to the evidence.*
PON	*Positive Or Negative, the result of a verification process.*
TSA	*Trusted time Stamping Authority.*
TST	*Time Stamping Token generated by the TSA.*
TTP	*Trusted Third Party.*

5 Requirements

5.1 Two entities wishing to use one of the mechanisms specified in this part of ISO/IEC 13888 must both trust the same third party.

5.2 Prior to the use of these mechanisms, it is assumed that each entity shares a secret key with the trusted third party. The trusted third party *(TTP)* also holds a single key known only to itself.

NOTE – Key management, key generation and key establishment mechanisms are defined in the multipart standard ISO/IEC 11770.

5.3 A common function *Imp* is shared by all entities in the non-repudiation service. The function *Imp* shall be either the identity function or a collision-resistant hash-function as defined in ISO/IEC 10118.

5.4 A function *MAC* chosen for envelope *(SENV)* creation must be held by all participants in the non-repudiation service.

5.5 The *TTP* generating the non-repudiation tokens shall be able to access the time and date.

5.6 The strength of the mechanisms specified in this part of 13888 is dependent on the length and the secrecy of the key, on the nature of the function *MAC*, and on the length of the check value. These parameters shall be chosen to meet the required security level, as may be specified by the security policy.

6 Organization of this part of ISO/IEC 13888

The mechanisms described in this part of ISO/IEC 13888 require that each of the two entities involved are able to communicate separately with the *TTP*. They require the use of secure envelopes described in clause 7. The basic concepts for the generation and verification of non-repudiation tokens, and thus of evidence, are described in clause 8. The mechanisms described in clause 9 require the use of a *TTP* which needs to be called for every evidence generation and every evidence verification. Three variants of this mechanism are further described as examples in clause 10.

7 Secure envelopes

Two entities that share a secret key (known only to them) may send messages to one another using a method for data integrity known as a Secure ENVelope, *SENV*. A *SENV* is formed by protecting the input data items with a secret key. A *SENV* can also be used by a *TTP* for generating and verifying evidence, using the secret key held only by the *TTP*.

The method of creating a secure envelope is through the use of symmetric integrity techniques. The secret key *x* of entity *X* is used to compute a cryptographic check value $MAC_X(y)$ which is appended to the data *y:*

$$SENV_X(y) = y \,||\, MAC_X(y),$$

where $MAC_X(y)$ can be the message authentication code as specified in ISO/IEC 9797.

The function *MAC* shall fulfill the following requirements as specified in ISO/IEC 9798-4:

- for any key x and data string y, it shall be practical to compute $MAC_X(y)$;

- for any fixed key x, and given no prior knowledge of x, it shall be computationally infeasible to find a new pair (y',z) such that $MAC_X(y')=z$; even given knowledge of a set of pairs $(y_i\ ,\ z_i)$ such that $MAC_X(y_i)= z_i$ $(i=1,2,...)$, where the value of y_j may have been chosen after observing the value of z_i $(j=1,2,...,i-1)$.

8 Generation and verification of non-repudiation tokens

In the non-repudiation mechanisms described in this clause, the *TTP* acts as an evidence generation and an evidence verification authority. It is trusted to maintain the integrity of certain records and is directly involved in the resolution of any dispute.

8.1 Creation of tokens by the *TTP*

The *TTP* issues "tokens" to be associated with a message *m*. A token is a secure envelope formed by the *TTP* using its secret key on data specific to a message. Because no other entity knows secret key *ttp*, the *TTP* is the only one that can create or verify tokens. In ISO/IEC 13888-1 the generic non-repudiation token *(GNRT)* is defined as follows:

$$GNRT = text \,||\, SENV_X(y).$$

The *TTP* should check also the data items present in the evidence request before the tokens are issued.

8.2 Data items used in the non-repudiation mechanisms

8.2.1 Data items used in secure envelopes

The following data fields will form the contents of secure envelopes

$$SENV_X(z) = z \,||\, MAC_X(z)$$

to be exchanged during the non-repudiation mechanisms described in this part of ISO/IEC 13888:

$$z = Pol \,||\, f_i \,||\, A \,||\, B \,||\, C \,||\, D \,||\, E \,||\, T_g \,||\, T_i \,||\, Q \,||\, Imp(m).$$

The data field z consists of the following data items:

Pol the distinguishing identifier of the non-repudiation policy (or policies) which apply to the evidence,

f_i the type of non-repudiation service being provided,

A the distinguishing identifier of the originating entity,

B the distinguishing identifier of the entity interacting with the originating entity,

C the distinguishing identifier of the evidence generator,

D the distinguishing identifier of the evidence requester when different from the originating entity,

E the distinguishing identifiers of other entities involved with the action,

T_g the date and time the evidence was generated,

T_i the date and time the event or action took place,

Q optional data that need to be protected,

Imp(m) the imprint of a message *m* related to the action (either the hash-code of the message *m* or the message *m* itself).

NOTE – Depending on the non-repudiation policy, some data items may be optional.

8.2.2 Data items used in non-repudiation tokens

Non-repudiation tokens contain a text field denoted by *text:*

$$Non\text{-}repudiation\ token = text \,||\, SENV_{TTP}(z)$$

text includes additional data (like a message identifier or key identifier) that does not need to be cryptographically protected but may be needed to identify the message and the key used in the computation of the integrity check value *MAC.* This information depends upon the technique being used.

8.3 Non-repudiation tokens

Evidence is provided by non-repudiation tokens, and, if the policy requires it, by additional tokens such as a time stamping token (*TST*), and a token provided by another trusted fourth party (e.g., a Notary) giving additional assurance about an event or action, or about the existence of a message.

If the trusted third party is able to generate a trusted time stamp itself, the addition of a time stamping token (*TST*) as evidence is unnecessary. The time included in non-repudiation tokens (*NROT*, *NRDT*, *NRST* and *NRTT*) is regarded secure as it was provided by a trusted authority.

If the trusted third parties (*TTP*, *DA*) are unable to provide a trusted time stamp, then a time stamping token (*TST*) provided by the trusted time stamping authority (*TSA*) shall be added to the set of non-repudiation information to complete the evidence.

8.3.1 Non-repudiation of origin token

A non- repudiation of origin token (*NROT*) is created by the *TTP* at the request of the originator:

$NROT = text \,||\, z_1 \,||\, MAC_{TTP}(z_1)$, with

$$z_1 = Pol \,||\, f_1 \,||\, A \,||\, B \,||\, C \,||\, D \,||\, T_g \,||\, Q \,||\, Imp(m).$$

The information z_1 necessary for the *NROT* consists of the following data items:

Pol the distinguishing identifier of the non-repudiation policy (or policies) which apply to the evidence,

f_1 a flag indicating non-repudiation of origin,

A the distinguishing identifier of the originator,

B the distinguishing identifier of the intended recipient,

C the distinguishing identifier of the *TTP* generating the evidence,

D the distinguishing identifier of the observer, if an independent observer is involved,

T_g the date and time the evidence was generated,

Q optional data that need to be protected,

Imp(m) the imprint of the message *m*.

8.3.2 Non-repudiation of delivery token

A non-repudiation of delivery token (*NRDT*) is created by the *TTP* at the request of the recipient:

$NRDT = text \,||\, z_2 \,||\, MAC_{TTP}(z_2)$, with

$z_2 = Pol \,||\, f_2 \,||\, A \,||\, B \,||\, C \,||\, D \,||\, T_g \,||\, T_2 \,||\, Q \,||\, Imp(m).$

The information z_2 necessary for the *NRDT* consists of the following data items:

Pol the distinguishing identifier of the non-repudiation policy (or policies) which apply to the evidence,

f_2 a flag indicating non-repudiation of delivery,

A the distinguishing identifier of the originator,

B the distinguishing identifier of the recipient,

C the distinguishing identifier of the evidence generator,

D the distinguishing identifier of the observer, if an independent observer is involved,

T_g the date and time the evidence was generated,

T_2 the date and time the message was delivered.

Q optional data that need to be protected,

Imp(m) the imprint of the message *m*.

8.3.3 Non-repudiation of submission token

The non-repudiation of submission token (*NRST*) is generated by the delivery authority (*DA*). The delivery authority, a trusted third party, may be the same as the one generating the *NROT* or *NRDT*.

$NRST = text \,||\, z_3 \,||\, MAC_{DA}(z_3)$ with

$z_3 = Pol \,||\, f_3 \,||\, A \,||\, B \,||\, C \,||\, D \,||\, E \,||\, T_g \,||\, T_3 \,||\, Q \,||\, Imp(m).$

The information z_3 necessary for the *NRST* consists of the following data items:

Pol the distinguishing identifier of the non-repudiation policy (or policies) which apply to the evidence,

f_3 a flag indicating non-repudiation of submission,

A the distinguishing identifier of the originator (submitting entity),

B the distinguishing identifier of the intended recipient,

C the distinguishing identifier of the delivery authority (*DA*),

D the distinguishing identifier of the observer, if an independent observer is involved,

E the distinguishing identifier of the authority acting on behalf of the delivery authority (optional),

T_g the date and time the evidence was generated,

T_3 the date and time the message was submitted for transport,

Q optional data that need to be protected,

Imp(m) the imprint of the message submitted for delivery.

8.3.4 Non-repudiation of transport token

The non-repudiation of transport token (*NRTT*) is generated by a delivery authority *DA*:

$NRTT = text \,||\, z_4 \,||\, MAC_{DA}(z_4)$ with

$z_4 = Pol \,||\, f_4 \,||\, A \,||\, B \,||\, C \,||\, D \,||\, E \,||\, T_g \,||\, T_4 \,||\, Q \,||\, Imp(m).$

The information z_4 necessary for the *NRTT* consists of the following data items:

Pol the distinguishing identifier of the non-repudiation policy (or policies) which apply to the evidence,

f_4 a flag indicating non-repudiation of transport,

A the distinguishing identifier of the originator,

B the distinguishing identifier of the recipient,

C the distinguishing identifier of the delivery authority,

D the distinguishing identifier of the observer, if an independent observer is involved,

E the distinguishing identifier of the authority acting on behalf of the delivery authority (optional),

T_g the date and time the evidence was generated,

T_4 the date and time the message was delivered to the data storage of the recipient,

Q optional data that need to be protected,

Imp(m) the imprint of the message *m*.

8.3.5 Time stamping token

The time stamping token (*TST*) provided by the time stamping authority (*TSA*) is defined as follows:

$TST = text \,||\, z_5 \,||\, MAC_{TSA}(z_5)$ with

$z_5 = Pol \,||\, f_5 \,||\, TSA \,||\, T_g \,||\, Q \,||\, Imp(m).$

The data field z_5 consists of the following data items:

Pol the distinguishing identifier of the non-repudiation policy (or policies) which apply to the evidence,

f_5 a flag indicating time stamping token,

TSA the distinguishing identifier of the Time Stamping Authority,

T_g the date and time when the evidence (i.e., *TST*) is generated for a particular message,

Q optional data that need to be protected,

Imp(m) the imprint of the message *m* with which a time stamp is to be associated.

8.4 Verification of tokens by the *TTP*

At some point during the non-repudiation exchange, it may be necessary for the *TTP* to verify tokens (as defined above) received from an entity. It may also be necessary to re-verify the tokens some time after the exchange is completed, or to provide evidence as to their veracity to some fourth party.

The process of verification includes not only checking that a token was created by the *TTP*, but also that the token is appropriately associated with the data field of the message for which it was created. To check that a token was created for a given message, an entity can verify the message by comparing the *Imp(m)* computed from the message and the *Imp(m)* contained in the data field *z*, and then request the *TTP* to verify the token together with its data field.

To verify secure envelopes generated by using symmetric integrity techniques, the verification operation applied consists of recalculating the cryptographic check value $MAC_X(y)$ using the appropriate secret key *x* of entity *X* and data y contained in the secure envelope, and of comparing this result with that presented.

Two methods of verification of tokens by the *TTP* are provided.

8.4.1 On-line verification of the token

In this method, the *TTP* uses a security module containing the secret key *ttp* to verify the token. The security module compares the token with a value that is internally regenerated using the data item z_i and the secret key *ttp*, and returns the outcome of the comparison by specifying whether the token is valid or not. Since the key *ttp* is not known by anyone else than the *TTP*, the token presented for verification is considered authentic, if the security module returns that the token is valid.

8.4.2 Table of tokens

In this method, a table of all tokens issued by the *TTP* is stored. For each token created, the *TTP* records the token along with its associated data field (z_i) and the key identifier of the secret key *ttp.* To verify it, the *TTP* uses the token as an index into the table to look it up. If the token presented for verification is found in the table, and the data field that is presented with the token (or, is part of the token) corresponds to the data field associated with it in the table, then the token is considered to be authentic.

9 Specific non-repudiation mechanisms

The non-repudiation mechanisms in this clause allow for generation of evidence for non-repudiation of origin (*NRO*), delivery (*NRD*), submission (*NRS*), and of transport (*NRT*). In addition the mechanism for generating the time stamp is defined. Entity A wishes to send a message *m* to entity B and thus will be the originator of the non-repudiation transfer. Entity B will be the recipient.

In some mechanisms described in Clause 9, the z_i data field of the request does not contain time information. Such time information will be provided by the *TTP* (or *DA*) or by the time stamping authority *TSA* upon request of the *TTP* (or *DA*).

NOTE – In case that *Imp(m)* is the message *m*, it is not necessary to send *m* together with tokens, and the steps for verifying *Imp(m)* are also omitted.

9.1 Mechanism for non-repudiation of origin

The originator has created a message which he will send to a specified recipient. The recipient can check that this message is from the claimed sender by using the *TTP* to verify the associated non-repudiation of origin token.

In the first transaction of this mechanism, the originator forms the data and transmits it in a *SENV* to the *TTP*. The *TTP* generates the non-repudiation of origin token (*NROT*) and returns this to the originator *A*. In the second transaction, the *NROT* concatenated to message *m* is sent from the originator *A* to the recipient *B.* In the third transaction, the recipient sends the *NROT* enclosed in a secure envelope to the *TTP* for verification. Non-repudiation of origin is established in the third transaction.

9.1.1 Transaction 1 – between originator *A* and *TTP*

a. Entity *A* generates a secure envelope $SENV_A(z'_1)$ using key *a*, where z'_1 is the z_1 specified in clause 8.3.1 with the data item T_g being empty. Entity *A* then requests an *NROT* by sending the secure envelope to the *TTP*.

b. The *TTP* verifies that the secure envelope is from entity *A*. If it is, the *TTP* then completes z_1 by inserting the data item T_g and computes the

$NROT = text \; || \; z_1 \; || \; MAC_{TTP}(z_1)$

using key *ttp* and returns $SENV_A(NROT)$ to *A*.

c. Entity *A* verifies that $SENV_A(NROT)$ is from the *TTP*.

9.1.2 Transaction 2 – from originator *A* to recipient *B*

Entity *A* sends to *B:* *m* || *NROT.*

9.1.3 Transaction 3 – between recipient *B* and *TTP*

a. Entity *B* verifies the value of *Imp(m)* contained in z_1, then generates $SENV_B(NROT)$ using key *b* and sends it to the *TTP* in order to request verification of the *NROT* received from *A*.

b. The *TTP* verifies that $SENV_B(NROT)$ is from *B* and also verifies that the *NROT* is authentic. If $SENV_B(NROT)$ is not valid the mechanism is to be terminated. If $SENV_B(NROT)$ is valid, the *TTP* sends $SENV_B(PON \; || \; NROT)$ to *B* where *PON* is positive if the *NROT* is authentic and negative if the *NROT* is not authentic.

c. Entity *B* verifies that $SENV_B(PON \; || \; NROT)$ is from the *TTP*. If it is valid and the verification is positive, non-repudiation of origin (i.e., the message came from *A*) is established.

d. The *NROT* is saved for future non-repudiation of origin.

9.2 Mechanism for non-repudiation of delivery

After receipt of the message *m*, entity *B* sends in the first transaction of this mechanism, a request to generate a non-repudiation of delivery token to the *TTP* enclosed in a secure envelope. The *TTP* generates the non-repudiation of delivery token (*NRDT*) and returns this to the recipient *B*. In the second transaction, the *NRDT* is sent by the recipient *B* to the originator *A.* In the third transaction, the originator sends the *NRDT* enclosed in a secure envelope to the *TTP* for verification. Non-repudiation of delivery is established in the third transaction.

9.2.1 Transaction 1 – between recipient *B* and *TTP*

a. Entity *B* generates a secure envelope $SENV_B(z'_2)$ using key *b*, where z'_2 is the z_2 specified in clause 8.3.2. with the data item T_g being empty. Entity *B* then requests an *NRDT* by sending the secure envelope to the *TTP*.

b. The *TTP* verifies that the secure envelope is from entity *B*. If it is, the *TTP* completes z_2 by inserting the data item T_g and computes

NRDT = $text \parallel z_2 \parallel MAC_{TTP}(z_2)$ using key *ttp*

c. Entity *B* verifies that $SENV_B(NRDT)$ is from the *TTP*.

9.2.2 Transaction 2 – from recipient *B* to originator *A*

Entity *B* sends to *A*: *NRDT.*

9.2.3 Transaction 3 – between originator *A* and *TTP*

a. Entity *A* verifies the value of *Imp*(*m*) contained in z_2, then generates $SENV_A(NRDT)$ using key *a* and sends it to the *TTP* in order to request verification of the *NRDT* received from *B.*

b. The *TTP* verifies that $SENV_A(NRDT)$ is from *A* and also verifies that the *NRDT* is authentic. If $SENV_A(NRDT)$ is not valid the mechanism is to be terminated. If $SENV_A(NRDT)$ is valid, the *TTP* sends $SENV_A(PON \parallel NRDT)$ to *A*, where *PON* is positive if the *NRDT* is authentic and negative if the *NRDT* is not authentic.

c. Entity *A* verifies that $SENV_A(PON \parallel NRDT)$ is from the *TTP*. If it is valid and the verification is positive, non-repudiation of delivery is established.

d. The *NRDT* is saved for future non-repudiation of delivery.

9.3 Mechanism for non-repudiation of submission

In the first transaction of this mechanism, a submitting entity *X* sends a message *m* to a delivery authority *DA* for onward delivery. In the second transaction, the non-repudiation of submission token (*NRST*) is sent from the delivery authority *DA* to the entity *X.* The non-repudiation of submission is established in the second transaction.

9.3.1 Transaction 1 – from submitting entity *X* to delivery authority *DA*

a. Entity *X* generates a secure envelope $SENV_X(z'_3)$ using key *x*, where z'_3 is the z_3 specified in clause 8.3.3 with the data item T_g being empty. Entity *X* then requests an *NRST* by sending the secure envelope together with *m* to the *DA*.

b. The *DA* verifies that the secure envelope is from entity *X* and that the message *m* is valid by checking *Imp*(*m*). If both are valid, the *DA* completes z_3 by inserting the data item T_g and then computes the

NRST = $text \parallel z_3 \parallel MAC_{DA}(z_3)$ using key *da***.**

9.3.2 Transaction 2 – from delivery authority *DA* to entity *X*

a. *DA* returns $SENV_X(NRST)$ to the submitting entity *X*.

b. Entity *X* verifies that $SENV_X(NRST)$ is from the *DA*.

c. If it is valid, the *NRST* is saved as evidence for non-repudiation of submission (i.e., the message was submitted).

9.4 Mechanism for non-repudiation of transport

In the first transaction of this mechanism, a sending entity *X* sends message *m* to delivery authority *DA* for onward delivery. In the second transaction, the message *m* is sent by delivery authority *DA* to a receiving entity *Y*. In the third transaction, the non-repudiation of transport token (*NRTT*) is generated by delivery authority *DA* and sent to entity *X,* the originating entity of the message *m.* The non-repudiation of transport is established in the third transaction.

9.4.1 Transaction 1 – from entity X to delivery authority *DA*

Entity *X* sends the message *m* and a request for *NRTT* token to the delivery authority *DA.*

9.4.2 Transaction 2 – from delivery authority *DA* to entity *Y*

Delivery authority *DA* sends the message *m* to Entity *Y.*

9.4.3 Transaction 3 – from delivery authority *DA* to entity *X*

a. Delivery authority *DA* forms the *NRTT*:

$NRTT = text \parallel z_4 \parallel MAC_{DA}(z_4)$, using the key *da*, where z_4 is specified in clause 8.3.4.

b. Delivery Authority *DA* sends the $SENV_X(NRTT)$ to entity *X*.

c. Entity *X* checks the $SENV_X(NRTT)$ and its content. If it is valid, the *NRTT* is saved as the evidence for non-repudiation of transport (i.e., the message was delivered to the intended recipient *Y*).

9.5 Mechanism for obtaining a time stamping token

The time stamping token (*TST*) is generated by a trusted time stamping authority (*TSA*) on request of the entity *X.*

In the first transaction the requesting entity *X* sends a message z'_5 that is completed by the *TSA* with time T_g. In the second transaction, the *TSA* sends back to the requesting entity the time stamping token (*TST*).

9.5.1 Transaction 1 – from entity *X* to time stamping authority *TSA*

a. Entity *X* generates a secure envelope $SENV_X(z'_5)$ using key x, where z'_5 is a subset of z_5 defined in clause 8.3.5 with the data item T_g being empty. Entity *X* then requests a time stamping token by sending the secure envelope to the *TSA*.

b. *TSA* generates the T_g consisting of date and time data, and thus completing the data field z'$_5$ to z$_5$.

c. *TSA* generates *TST*:

TST= text || z$_5$ || $MAC_{TSA}(z_5)$.

9.5.2 Transaction 2 – from time stamping authority *TSA* to entity *X*

a. The time stamping authority *TSA* returns the time stamping token (*TST*) to the requesting entity *X* in a secure envelope $SENV_X(TST)$ using key *x* known to entity *X* and to *TSA.*

b. Entity *X* checks the validity of the secure envelope.

10 Examples of non-repudiation mechanisms

The non-repudiation mechanisms in this clause provide non-repudiation of origin and non-repudiation of delivery between entities *A* and *B*. Entity *A* wishes to send a message to entity *B* and thus will be the originator of the non-repudiation exchange. Entity *B*, as the message recipient, will be the recipient. Prior to use of a mechanism, it is assumed that keys *a* and *b* are in place at entity *A* and entity *B,* respectively, and that the *TTP* possesses keys *a* and *b* in addition to its own key *ttp.*

Three different mechanisms (M1, M2, and M3) for non-repudiation using an on-line *TTP* are provided.

NOTES

1. By letting the *SENV* messages also include time stamps or sequence numbers, protection against unauthorized delay or replay of messages can be achieved. By letting the *NROT* and *NRDT* include time stamps, future verification of the time stamps at which a message was transferred can be obtained.

2. In case that *Imp(m*) is the message *m*, it is not necessary to send *m* together with tokens, and the steps for verifying *Imp(m)* are also omitted.

10.1 Mechanism M1: Mandatory *NRO*, optional *NRD*

Non-repudiation of origin is established in three transactions between the entities and the *TTP*. If the optional *NRD* steps are continued (at the recipient's prerogative), non-repudiation of delivery is established within two more transactions (see Figure 1).

NOTE – While it is up to the recipient to continue the steps required for non-repudiation of delivery, it is important to note that this optional non-repudiation of delivery is fully binding once it has been established.

10.1.1 Transaction 1 – between originator *A* and *TTP*

a. Entity *A* generates a secure envelope $SENV_A(z'_1)$ using key *a* , where z'_1 is the z_1 specified in clause 8.3.1. with the data item T_g being empty. Entity *A* then requests an *NROT* by sending the secure envelope to the *TTP*.

b. The *TTP* verifies that the secure envelope is from entity *A*. If it is, the *TTP* completes z_1 by inserting the data item T_g and computes the

NROT = text || z_1 || $MAC_{TTP}(z_1)$ using key *ttp* and returns $SENV_A(NROT)$ to *A*.

c. Entity *A* verifies that $SENV_A(NROT)$ is from the *TTP*.

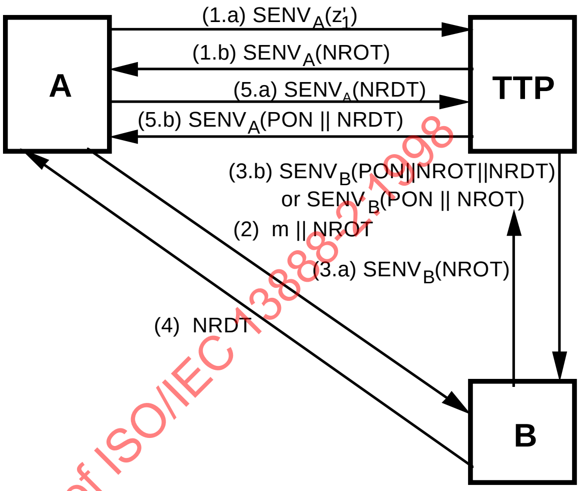

Figure 1 – Mechanism M1

10.1.2 Transaction 2 – from originator *A* to recipient *B*

Entity *A* sends to *B:* *m* || *NROT*.

10.1.3 Transaction 3 – between recipient *B* and *TTP*

a. Entity *B* verifies the value of *Imp(m*) contained in z_1, then generates $SENV_B(NROT)$ using key *b* and sends it to the *TTP* in order to request verification of the *NROT* received from *A*.

b. The *TTP* checks $SENV_B(NROT)$ and *NROT*. If both are valid, the *TTP* generates the non-repudiation of delivery token *NRDT* and sends $SENV_B(PON$ || *NROT* || *NRDT),* where *PON* is positive, to *B*. If the *SENV* is valid, but the *NROT* is not, the *TTP* sends $SENV_B(PON$ || *NROT),* where *PON* is negative, to *B*.

c. Entity *B* verifies that $SENV_B(PON$ || *NROT* ||*NRDT)* is from the *TTP*. If it is valid and *PON* is positive, non-repudiation of origin (i.e., the message came from *A*) is established. Alternatively, if *B* receives $SENV_B(PON$ || *NROT)* and *PON* is negative*,* then the *NROT* is not valid and the mechanism is to be terminated.

d. The *NROT* is saved for future non-repudiation of origin.

10.1.4 Transaction 4 – from recipient *B* to originator *A*

Entity *B* sends the *NRDT* to *A*.

10.1.5 Transaction 5 – between originator *A* and TTP

a. Entity *A* verifies the value of *Imp(m)* contained in z_2, then generates $SENV_A(NRDT)$ using key *a* and sends it to the *TTP* in order to request verification of the *NRDT* received from *B.*

b. The *TTP* verifies that $SENV_A(NRDT)$ is from *A* and also verifies that the *NRDT* is authentic. If both are valid, the *TTP* sends $SENV_A(PON \| NRDT)$, where PON is positive, to *A*. If the *NRDT* is not valid, the *TTP* sends $SENV_A(PON \| NRDT)$, where *PON* is negative, to *A.*

c. Entity *A* verifies that $SENV_A(PON \| NRDT)$ is from the *TTP*. If it is valid and the verification is positive, non-repudiation of delivery is established.

d. The *NRDT* is saved for future non-repudiation of delivery.

10.2 Mechanism M2: Mandatory *NRO*, mandatory *NRD*

Non-repudiation of origin and non-repudiation of delivery are established in four transactions between the two entities and the *TTP*. In this mechanism, the *TTP* sends the message receipt directly to *A* in a *SENV* at the same time that he sends it to *B* (see Figure 2).

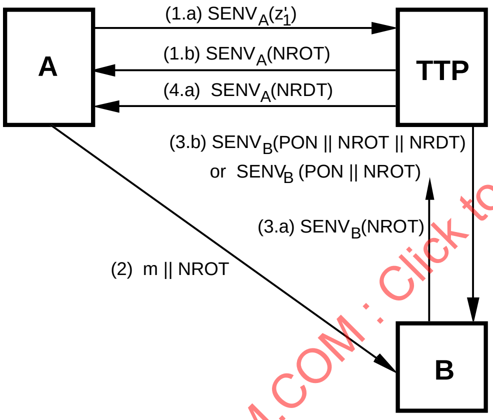

Figure 2 – Mechanism M2

10.2.1 Transaction 1 – between originator *A* and *TTP*

a. Entity *A* generates a secure envelope $SENV_A(z'_1)$ using key *a*, where z'_1 is the z_1 specified in clause 8.3.1. with the data item T_g being empty. Entity *A* then requests an *NROT* by sending the secure envelope to the *TTP*.

b. The *TTP* verifies that the secure envelope is from entity *A*. If it is, the *TTP* completes z_1 by inserting the data item T_g and then computes the $NROT = text \| z_1 \| MAC_{TTP}(z_1)$ using key *ttp* and returns $SENV_A(NROT)$ to *A* using key *a.*

c. Entity *A* verifies that $SENV_A(NROT)$ is from the *TTP.*

10.2.2 Transaction 2 – from originator *A* to recipient *B*

Entity *A* sends to *B:* $m \| NROT$.

10.2.3 Transaction 3 – between recipient *B* and *TTP*

a. Entity *B* verifies the value of *Imp(m)* contained in z_1, then generates $SENV_B(NROT)$ using key *b* and sends it to the *TTP* in order to request verification of the *NROT* received from *A*.

b. The *TTP* verifies that $SENV_B(NROT)$ is from *B* and that the *NROT* is authentic. If both are valid, the *TTP* generates the *NRDT* and sends

$$SENV_B(PON \| NROT \| NRDT),$$

where *PON* is positive, to *B*. If the *SENV* is valid, but the *NROT* is not, the *TTP* sends

$$SENV_B(PON \| NROT),$$

where *PON* is negative, to *B*.

c. Entity *B* verifies that $SENV_B(PON \| NROT \| NRDT)$ is from the *TTP*. If it is valid and *PON* is positive, non-repudiation of origin is established. Alternatively, if *B* receives $SENV_B(PON \| NROT)$ and *PON* is negative, then the *NROT* is not valid and the mechanism is to be terminated.

d. The *NROT* is saved for future non-repudiation of origin.

10.2.4 Transaction 4 – between *TTP* and originator *A*

a. Immediately after sending the *NRDT* to *B* in Transaction 3, the *TTP* also sends $SENV_A(NRDT)$ to *A*.

b. Entity *A* checks $SENV_A(NRDT)$ and *NRDT.* If both are valid, non-repudiation of delivery (i.e., the message was received by *B)* is established.

c. The *NRDT* is saved for future non-repudiation of delivery.

10.3 Mechanism M3: Mandatory *NRO* and *NRD* with intermediary *TTP*

Non-repudiation of origin and non-repudiation of delivery are established in four transactions between the two entities and the *TTP*. In Mechanism M3, the *TTP* acts as an intermediary between the originator and the recipient – the two entities never correspond directly. To accomplish this, entity A sends the message to the *TTP* as part of Transaction 1, and the *TTP* passes it to entity *B* as part of Transaction 2 (see Figure 3).

As the *TTP* in this mechanism takes the role of a delivery authority it may optionally generate and send non-repudiation of submission and non-repudiation of transport tokens to the originating entity.

10.3.1 Transaction 1 – between originator *A* and *TTP*

a. Entity *A* generates a secure envelope $SENV_A(z'_1)$ using key *a*, where z'_1 is the z_1 specified in clause

8.3.1 with the data item T_g being empty. Entity *A* then requests an *NROT* by sending the secure envelope together with the message *m* to the *TTP*.

b. The *TTP* verifies that the secure envelope is from entity *A*. If it is, the *TTP* completes z_1 by inserting the data item T_g and computes the $NROT = text \,||\, z_1 \,||\, MAC_{TTP}(z_1)$ using key *ttp* and returns $SENV_A(NROT)$ to *A* using key *a*.

c. Entity *A* verifies that $SENV_A(NROT)$ is from the *TTP*.

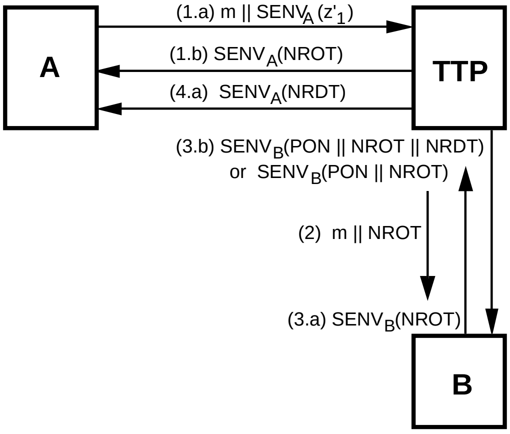

Figure 3 – Mechanism M3

10.3.2 Transaction 2 – from *TTP* to recipient *B*

a. The *TTP* sends *m* and the *NROT* to *B.*

10.3.3. Transaction 3 – between recipient *B* and *TTP*

a. Since the *NROT* was not received in a secure envelope, *B* must verify it with the *TTP*, so he verifies *Imp(m)* and sends $SENV_B(NROT)$ to the *TTP*.

b. The *TTP* verifies that $SENV_B(NROT)$ is from *B* and also verifies the authenticity of the *NROT.* If both are valid, the *TTP* creates the non-repudiation of delivery token *NRDT* and responds to *B* with *PON* positive by sending $SENV_B(PON \,||\, NROT \,||\, NRDT)$. If the *SENV* is valid, but the *NROT* is not, the *TTP* sends $SENV_B(PON \,||\, NROT)$ to *B*, where *PON* is negative.

c. *B* verifies that the *SENV* is from the *TTP*. If it is valid and *PON* is positive, non-repudiation of origin is established. Alternatively, if *B* receives $SENV_B(PON \,||\, NROT)$ and *PON* is negative, then the *NROT* is not valid and the mechanism is to be terminated.

d. The *NROT* is saved for future non-repudiation of origin.

10.3.4 Transaction 4 – between *TTP* and originator *A*

a. Immediately after sending the *NRDT* to *B* in Transaction 3, the *TTP* also sends it to *A*, $SENV_A(NRDT)$.

b. *A* verifies that this was received from the *TTP*, and non-repudiation of delivery is established.

c. The *NRDT* is saved for future non-repudiation of delivery.