International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1156

(06/2013)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Secure applications and services – Security protocols

Non-repudiation framework based on a one-time password

Recommendation ITU-T X.1156

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols	**X.1150–X.1159**
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1339
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1156

Non-repudiation framework based on a one-time password

Summary

Recommendation ITU-T X.1156 provides a non-repudiation framework based on a one-time password (OTP) to enhance trust between transaction entities. In addition, this Recommendation describes the security requirements of OTP-based non-repudiation services, as well as the mechanisms for generating non-repudiation tokens. The originator of the transactions may request the trusted third party (TTP) to generate the non-repudiation token and the recipient may request the TTP to generate the non-repudiation of delivery token. Both the originator and the recipient may request TTP to verify the non-repudiation tokens.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T X.1156	2013-06-13	17

Keywords

Non-repudiation framework, one-time password, trusted third party.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Page

Recommendation ITU-T X.1156

Non-repudiation framework based on a one-time password

1 Scope

Recommendation ITU-T X.1156 provides a non-repudiation framework based on a one-time password (OTP) to provide trust mechanisms between transaction entities. Also, this Recommendation describes the security requirements of a one-time password based on a non-repudiation service, as well as the mechanisms for generating non-repudiation tokens.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.813] Recommendation ITU-T X.813 (1996) | ISO/IEC 10181-4:1997, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Non-repudiation framework*.

[ITU-T X.1153] Recommendation ITU-T X.1153 (2011), *Management framework of a one time password-based authentication service*.

[ISO/IEC 13888-2] ISO/IEC 13888-2:2010, *Information technology – Security techniques – Non-repudiation – Part 2: Mechanisms using symmetric techniques*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 authentication framework [ITU-T X.1153]: A general framework for the provision of authentication service; two party and (trusted) third party authentication frameworks (based on [b-ITU-T X.811]).

3.1.2 one-time password (OTP) [ITU-T X.1153]: A password that can be used only once as it is changed every time an OTP user logs into the computer system and network. It is secure against the passive attacks allowed by the replaying of captured reusable passwords such as traditional fixed passwords (based on [b-IETF RFC 2289]).

3.1.3 OTP service provider [ITU-T X.1153]: The provider(s) of the OTP validation service offering multi-factor authentication services to other service providers, e.g., Internet service provider or application/contents service provider. A trusted third party would be a good example for an OTP service provider to perform OTP validation more efficiently. The OTP service provider does not necessarily have the role of identity provider as defined in [b-ITU-T X.1141].

3.1.4 OTP token [ITU-T X.1153]: A physical device that generates an OTP, wherein a token means that the OTP user possesses and controls a key or password used to authenticate the OTP user's identity. Such a physical device embeds the display showing an OTP and the numeric keypad optionally.

3.1.5 OTP token identifier [ITU-T X.1153]: A unique identifier assigned to each OTP token for distinguishing it from other OTP tokens. It may consist of alphanumeric characters.

3.1.6 trusted third party [b-ITU-T X.810]: A security authority or its agent that is trusted with respect to some security-relevant activities (in the context of a security policy).

3.2 Terms defined in this Recommendation

This Recommendation defines the following term:

3.2.1 non-repudiation token: This token is the unforgeable evidence for the non-repudiation service.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

HMAC	Hash-based Message Authentication Code
MAC	Message Authentication Code
NRD	Non-Repudiation of Delivery
NRDT	Non-Repudiation of Delivery Token
NRO	Non-Repudiation of Origin
NROT	Non-Repudiation of Origin Token
NRT	Non-Repudiation Token
OSP	OTP Service Provider
OTP	One-Time Password
PON	Positive Or Negative
SENV	Secure Envelope
SEVM	Secure Envelope using MAC
SEVO	Secure Envelope using OTP
TSA	Time Stamping Authority
TTP	Trusted Third Party

5 Conventions

The keywords "**is required to**" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keywords "**is recommended**" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "**is prohibited from**" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keywords "**can optionally**" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with this Recommendation.

6 Overview of non-repudiation framework based on a one-time password

6.1 Introduction

The non-repudiation framework based on an OTP is to prevent entities from denying that they have sent or received electronic transaction data in the telecommunication network using an OTP. The framework supports the implementation and deployment of an OTP-based non-repudiation service, and defines entities, non-repudiation tokens and non-repudiation processes. Also, this framework provides security requirements of an OTP-based non-repudiation service, as well as mechanisms for generating non-repudiation tokens.

Since the framework may be provided through an OTP algorithm such as the techniques of symmetric, hash-based message authentication code (HMAC), etc., the framework could not be fairly enabled without an in-line trusted third party (TTP).

The non-repudiation process is composed of four distinct phases: evidence generation, evidence transfer, evidence verification, and dispute resolution, see [ITU-T X.813].

An originator of electronic transaction data generates an OTP using an OTP generation key in conjunction with the data, and sends the TTP (either directly or through a recipient) a request for some evidence of origin. Furthermore, a recipient may request the TTP to generate the evidence of delivery. Both pieces of evidence of the transaction data are called non-repudiation tokens. After the transaction, both the originator and the recipient may request the TTP to verify the non-repudiation tokens.

6.2 Entities and keys

The entities involved in the non-repudiation framework based on an OTP include an OTP user, a service provider and a TTP.

– OTP user: The OTP user, as the originator of the transaction, generates a secure envelope which is a request message of a non-repudiation of origin token using the OTP user's key. The OTP user can request a verification of the non-repudiation of delivery token by the TTP using the OTP user's key which is needed to resolve disputes.

– Service provider: The service provider, as the recipient of the transaction, generates a secure envelope which is a request message of a non-repudiation of delivery token using the service provider's key. The service provider can request a verification of the non-repudiation of origin token by the TTP using the service provider's key which is needed to resolve disputes.

– Trusted third party: The TTP generates and verifies non-repudiation tokens for other entities. Therefore, the TTP is required to be trusted by both the OTP user and by the service provider. An OTP service provider (OSP) in an OTP-based management service [ITU-T X.1153] would be a suitable candidate for being the TTP when the OSP is managed independently by both the OTP user and by the service provider.

The keys involved in the non-repudiation framework based on an OTP include an OTP user's key, a service provider's key and a TTP's key.

– OTP user's key: The OTP user and TTP are required to pre-share an OTP generation key which generates an OTP and maintains the integrity of the transactions between an OTP user and a TTP.

– Service provider's key: The service provider and TTP are required to pre-share a symmetric key or an OTP generation key which maintains the integrity of the transactions between the service provider and TTP.

- Trusted third party's key: The TTP is prohibited from sharing the TTP's key with any other entities. The TTP only can generate and verify non-repudiation tokens using the TTP's key. Thus, the OTP user and the service provider could request the TTP to generate and verify the non-repudiation tokens.

6.3 Secure envelope

In the telecommunication network, it is required to maintain the integrity of transaction data in the non-repudiation framework based on an OTP. Therefore, the framework uses two types of secure envelope (SENV) to protect the integrity of the data. One type uses the cryptographic check function of the message authentication code (MAC), and the other uses the OTP generation function. The MAC check function may be implemented by various data cryptographic algorithms, such as HMAC, encryption algorithm, and even asymmetric algorithm. The OTP generation function may be implemented by various standards, such as [b-IETF RFC 4226].

The framework defines SEVM as a secure envelope using the MAC check function and SEVO as a secure envelope using the OTP generation function. *X* is a pre-shared key which is the OTP user's key or the service provider's key. *y* is data to be securely enveloped.

The secure envelopes in the framework are defined as follows:

- $\mathrm{SEVM_x(y) = y \,||\, MAC_x(y)}$
- $\mathrm{SEVO_x(y) = y \,||\, OTP_x(y)}$

Alternatively, the framework also defines expression *z* instead of expression *y* only when the entities need to request an OTP-based non-repudiation token.

- $\mathrm{SEVM_x(z) = z \,||\, MAC_x(z)}$
- $\mathrm{SEVO_x(z) = z \,||\, OTP_x(z)}$

Expression *z* implies the request data for the non-repudiation token and consists of the following items:

- $z = I \,||\, T \,||\, A$

The expression *I* is an identifier data field of participating entities in the transaction, *T* is a time data field of date and/or time, *A* is an authentication data field for a non-repudiation service. These expressions defined in this Recommendation are also compatible with [ISO/IEC 13888-2]. The detailed description of each data field is explained in clauses 6.3.1, 6.3.2 and 6.3.3.

6.3.1 Identifier data field

Identifier data field *I* contains identifiers for a non-repudiation service, i.e., policy and identifiers of each entity.

- $I = Pol \,||\, A \,||\, B \,||\, C \,||\, D \,||\, E$

Identifier data field *I* consists of the following data items:

- *Pol:* identifiers of the OTP-based non-repudiation policy and a type of non-repudiation service;
- *A:* identifier of a message originator that defines the OTP user;
- *B:* identifier of a message recipient that defines the service provider;
- *C:* identifier of the evidence generator that defines the TTP;
- *D:* identifier of other entities involved with a non-repudiation service;
- *E:* identifiers of the OTP token (i.e., a serial number).

6.3.2 Time data field

Time data field *T* contains time information for a non-repudiation service. Time data for generating tokens T_g is required to be the trusted time such as a time stamp. If TTP cannot provide trusted time itself, it is necessary to have it provided by another trusted time stamping authority (TSA).

– $T = T_g \| T_i$

Time data field *T* consists of the following data items:

– T_g*:* a date and/or time when a non-repudiation token was generated,

– T_i*:* a date and/or time when a message was generated or delivered.

6.3.3 Authentication data field

Authentication data field *A* contains optional data and the imprint of a message for a non-repudiation service.

– $A = Q \| Imp(m)$

Authentication data field *A* consists of the following data items:

– *Q:* optional data that need to be protected,

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

7 Security requirements for an OTP-based non-repudiation service

This clause specifies the requirements that are satisfied by the OTP-based non-repudiation service. The use of an OTP-based non-repudiation service that complies with this Recommendation does not guarantee the security of the whole system. Moreover, the strength of the security of the non-repudiation service depends on the cryptographic algorithms that are used.

The general requirements for an OTP-based non-repudiation service are as follows:

– The OTP-based non-repudiation service requires an in-line TTP to generate and verify the non-repudiation tokens.

– The OTP-based non-repudiation service is required to use the algorithms approved by international standards organizations such as ITU, ISO/IEC, IETF, etc.

– The OTP-based non-repudiation service is required to generate the non-repudiation token based on trusted time. When a TTP cannot trust a local time, it is necessary to use an external time stamp service provided by a time stamping authority (TSA).

– The TTP of the OTP-based non-repudiation service is required to share the OTP user's key and the service provider's key only with the OTP user and the service provider, respectively, in a secure manner, but the TTP is prohibited from sharing the TTP's key with any other entity.

– The TTP providing the OTP-based non-repudiation service is required to keep the TTP's key valid until the expiration date of the non-repudiation tokens.

– The TTP providing the OTP-based non-repudiation service is required to limit the number of requests by successive OTP validation failures. Since the generated OTP is the same at the side of each entity and valid for a certain time, the TTP is required to count the guessing attacks of the OTP (e.g., if the OTP user fails to verify the SEVO ten times in one minute, the TTP blocks the OTP user's access).

– All entities of the OTP-based non-repudiation service are recommended to store and manage the non-repudiation tokens safely in secure storage.

– The OTP service provider that supports the OTP-based non-repudiation service is required to provide the long-term OTP validation operation for previously generated OTPs.

- The OTP device which supports the OTP-based non-repudiation service is recommended to provide a function that allows the transaction data to be submitted directly to the device.

8 Non-repudiation tokens

A non-repudiation token is the unforgeable evidence for the non-repudiation service. The token is generated by a TTP using the TTP's key. The TTP verifies a secure envelope that is a request message of a non-repudiation token received from an OTP user or service provider. If the secure envelope is valid, the TTP generates and sends the non-repudiation token. After transmitting the token, the non-repudiation token shall be stored by the OTP user and/or service provider and if one of the transaction entities claims repudiation of the transaction, the TTP will be able to validate the non-repudiation token.

The OTP-based non-repudiation token in the framework is generally defined as follows:

- $NRT = text \,||\, z \,||\, MAC_{ttp}(z)$, where $z = I \,||\, T \,||\, A$.

The OTP-based non-repudiation token establishes the accountability of entities involved in the service. The TTP generates an OTP-based non-repudiation token using the TTP's key *ttp*. The TTP shall generate a non-repudiation token using a trusted time and secure MAC algorithms. The text field *text* in a non-repudiation token includes additional information for a non-repudiation service.

8.1 OTP-based non-repudiation of origin token (NROT)

An OTP-based non-repudiation of origin token (NROT) is generated by the TTP at the request of the originator.

The NROT in the framework is defined as follows:

- $NROT = text \,||\, z \,||\, mac$

where:

$$z = I \,||\, T \,||\, A = (Pol \,||\, A \,||\, B \,||\, C \,||\, [D] \,||\, E)) \,||\, (T_g \,||\, T_i) \,||\, ([Q] \,||\, Imp(m))$$

and:

$$mac = MAC_{ttp}(z);\ [\,]\ \textit{optional input}$$

Pol of an identifier field *I* includes a non-repudiation policy and a flag for an OTP-based non-repudiation of origin (NRO). The remaining elements of identifier field *I* include an identifier of OTP user *A*, service provider *B*, TTP *C*, and OTP token *E*. If there is another entity D involved in this service, a token can optionally include the entity *D*. Time data field *T* includes the date and/or time of token generation T_g and request message generation T_i. Authentication data field *A* includes the protected data *Q* optionally and the imprint of the message *Imp(m)*.

8.2 OTP-based non-repudiation of delivery token (NRDT)

An OTP-based non-repudiation of delivery token (NRDT) is generated by the TTP at the request of the recipient.

The NRDT in the framework is defined as follows:

- $NRDT = text \,||\, z \,||\, mac$

where:

$$z = I \,||\, T \,||\, A = (Pol \,||\, A \,||\, B \,||\, C \,||\, [D] \,||\, E)) \,||\, (T_g \,||\, T_i) \,||\, ([Q] \,||\, Imp(m))$$

and:

$$mac = MAC_{ttp}(z);\ [\,]\ \textit{optional input}$$

Pol of an identifier field *I* includes a non-repudiation policy and a flag for an OTP-based non-repudiation of delivery (NRD). The remaining elements of identifier field *I* include an identifier of OTP user *A*, service provider *B*, TTP *C*, and OTP token *E*. If there is another entity involving this service, a token can include the entity *D* optionally. Time data field *T* includes the date and/or time of token generation T_g and message delivery T_i. Authentication data field *A* includes the protected data *Q* optionally and the imprint of the message *Imp(m)*.

9 Non-repudiation processes

Non-repudiation processes are composed of four distinct phases: evidence generation, evidence transfer, evidence verification and dispute resolution, see [ITU-T X.813]. This clause describes non-repudiation processes for both NRO and NRD.

9.1 Process of OTP-based non-repudiation of origin (NRO)

Non-repudiation of origin is intended to protect against the originator's false denial of having originated the message [b-ISO/IEC 13888-3]. OTP-based non-repudiation of origin provides a non-repudiation of origin service using a one-time password which is associated with electronic transaction data.

9.1.1 Evidence generation

The OTP user generates an SEVO and requests a token generation from the TTP. The TTP verifies the SEVO from the OTP user and then generates the NROT which is evidence data. An SEVO is able to be generated by the OTP user using a transaction data, time and/or date and the OTP user's key. Therefore, the OTP user can send the SEVO to the TTP directly or through a service provider.

a) The OTP user generates an OTP-based secure envelope $SEVO_X(z')$ using the OTP user's key *X*. *z'* is the *z* defined in clause 6.3 with time field T_g being empty. An OTP-based secure envelope is generated using an OTP generation function [ITU-T X.1153]. A one-time password is generated using synchronization data that is an input parameter for an OTP generation function inserting the *z* data field.

– $SEVO_X(z') = z' \parallel OTP_X(z')$

– $OTP_X(z')$ = *OTP generation function* (*X*, *Sync data*, [*Token activation data*]), where

Sync data = {*time and/or event counter and/or challenge or z*}; [] *optional input*

b) The TTP verifies the $SEVO_X(z')$ from the OTP user. If the secure envelope is valid, the TTP completes *z* by inserting a time field T_g. The TTP generates an NROT using the TTP's key *ttp*.

9.1.2 Evidence transfer

The TTP generates an SEVO using the OTP user's key for the generated NROT and sends the SEVO to the OTP user.

9.1.3 Evidence verification

If the service provider or another party wants to verify the NROT, the TTP is required to provide the method of token verification. There are two kinds of verification methods. For real-time verification, the TTP computes the NROT using the TTP's key *ttp* and then compares the result of computation with a received token. The TTP can use an SEVM or SEVO to request verification of the NROT. If the entity uses the SEVO, the TTP and entity are required to share the same OTP generation key in advance. If the entity uses the SEVM, the TTP and the entity share the symmetric key securely. In this framework, in particular, the OTP user shall use the SEVO for the request of a token verification.

a) The TTP receives the NROT which includes z^* and mac^* with additional information *text*. The z^* and mac^* are the received information of *z* and *mac,* respectively.

- $NROT = text \,||\, z^* \,||\, mac^*$

b) Generate $MAC_{ttp}\ (z^*)$ using TTP's key *ttp*.

c) Compare the received value mac^* and computed value $MAC_{ttp}\ (z^*)$. If they are same, the result of verification is "*valid*"; otherwise, the result of verification is "*invalid*".

An on-demand verification method is to verify a token for the entity's demand sometime after token issuance is complete.

9.1.4 Dispute resolution

Dispute of origin occurs when the OTP user denies having originated the message. This dispute can be resolved by the NROT. The service provider provides the NROT and transaction data (or an imprint of the message of the transaction data) to the TTP. The TTP verifies the NROT. If the NROT is valid, the transaction is valid. Otherwise, the transaction is not valid.

9.2 Process of OTP-based non-repudiation of delivery (NRD)

Non-repudiation of delivery is intended to provide protection against the recipient's false denial of having received and recognized the content of the message *m* [b-ISO/IEC 13888-3]. OTP-based non-repudiation of delivery (NRD) provides the non-repudiation of delivery service using a one-time password or a symmetric algorithm which is associated with electronic transaction data.

9.2.1 Evidence generation

The service provider generates the SEVO or SEVM and requests a token generation from the TTP. The TTP verifies the SEVO or SEVM from the service provider and then generates an NRDT which is evidence data. The SEVO can be generated by the service provider using transaction data, time and/or date and the service provider's key that is an OTP generation key.

a) The service provider generates an OTP-based secure envelope $SEVO_X(z')$ using the service provider's key *X* or a secure envelope $SEVM_X(z')$ using the service provider's key *X*. *z'* is the z defined in clause 6.3 with time field T_g being empty. The generating method of the SEVO is described in clause 9.1.1.

b) The TTP verifies the $SEVO_X(z')$ or $SEVM_X(z')$. If the secure envelope is valid, the TTP completes *z* by inserting a time field T_g. The TTP generates an NRDT using the TTP's key *ttp*.

9.2.2 Evidence transfer

The TTP generates an SEVO or SEVM using the service provider's key for the generated NRDT and sends the SEVO or SEVM to the service provider.

9.2.3 Evidence verification

If the OTP user (or another entity on behalf of the OTP user) wants to verify the NRDT, the TTP is required to provide the method of token verification. There are two kinds of verification methods. For real-time verification, the TTP computes an NRDT using the TTP's key *ttp* and then compares the result of computation with the received token. The TTP can use the SEVM or SEVO to request verification of an NRDT. If the entity uses an SEVO, the TTP and the entity shall share the OTP generation key in advance. Alternatively, if the entity uses an SEVM, the TTP and the entity share a symmetric key securely.

In this framework, in particular, the OTP user shall use an SEVO for the request of a token verification.

a) The TTP receives the NRDT which includes z^* and mac^* with additional information *text*. The z^* and mac^* are the received information of z and mac, respectively.

– $NROT = text \| z^* \| mac^*$

b) Generate $MAC_{ttp}(z^*)$ using TTP's key *ttp*.

c) Compare the received value mac^* and computed value $MAC_{ttp}(z^*)$. If they are the same, the result of verification is "*valid*"; otherwise, the result of verification is "*invalid*".

9.2.4 Dispute resolution

Dispute of delivery occurs when the service provider denies having received and recognized the content of the message. This dispute can be resolved by the NRDT. The OTP user provides the NRDT and transaction data (or imprint of message of the transaction data) to the TTP. The TTP verifies the NROT. If the NROT is valid, the transaction is valid; otherwise, the transaction is not valid.

10 Non-repudiation service models

The non-repudiation service models represent the relationship between the TTP and other entities. The TTP is the unique entity which can generate and verify the non-repudiation token in this framework. Thus, the OTP user and the service provider shall request the generation of the non-repudiation token by the TTP, and they will receive the token from the TTP.

10.1 Overview of the communication models

Since the non-repudiation token is to block a repudiation of the token requester's disputes, the actual beneficiary of the non-repudiation token is a corresponding entity, not the requesting entity. If the TTP sends the token to the corresponding entity directly, there is no need to verify the token. If the TTP responds with the token to the requesting entity, the requesting entity could deliver the token indirectly to the corresponding entity. In this case, an optional verification of the token can be done to check whether it has been generated by the TTP or not.

The following description provides the details of the communication models of the NROT and NRDT:

– **Communication model for direct NROT**: The non-repudiation of origin token (NROT) is to resolve disputes when the OTP user repudiates the transactions. Since the service provider receives an NROT from the TTP directly, the service provider could trust that the token is generated by the TTP (see Figure 1).

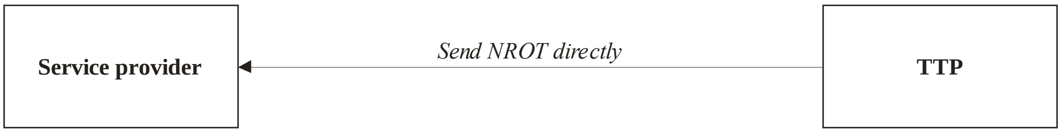

Figure 1 – Communication model for direct NROT

– **Communication model for direct NRDT**: The non-repudiation of delivery token (NRDT) is to resolve disputes when the service provider repudiates the transactions. Since the OTP user receives the NRDT from the TTP directly, the OTP user could trust that the token is generated by the TTP (see Figure 2).

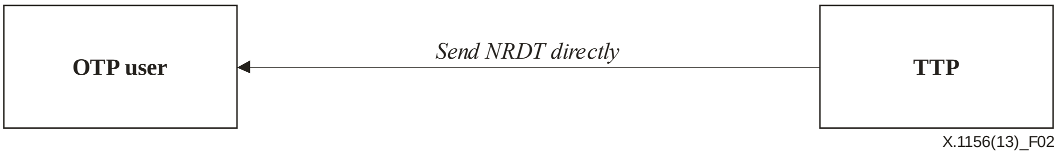

Figure 2 – Communication model for direct NRDT

- **Communication model for indirect NROT**: The service provider is the beneficiary of the NROT, and the OTP user delivers the NROT to the service provider on the TTP's behalf. Since the service provider receives the NROT from the OTP user, it is required to verify that the NROT is generated by the TTP. The verification of the NROT can optionally be performed offline after the transaction (see Figure 3).

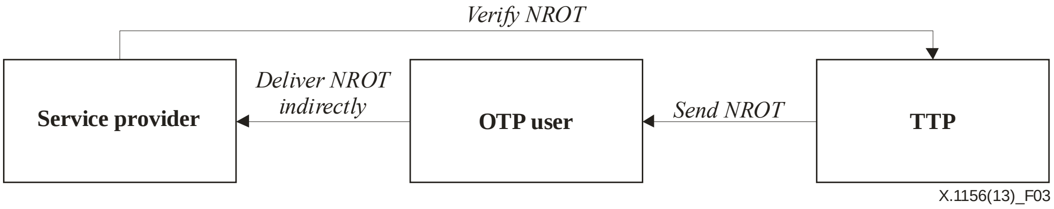

Figure 3 – Communication model for indirect NROT

- **Communication model for indirect NRDT**: The OTP user is the beneficiary of the NRDT, and the service provider delivers the NRDT to the OTP user on the TTP's behalf. Since the OTP user receives the NRDT from the service provider, it is required to verify that the NRDT is generated by the TTP. The verification of the NRDT can optionally be performed offline after the transaction (see Figure 4).

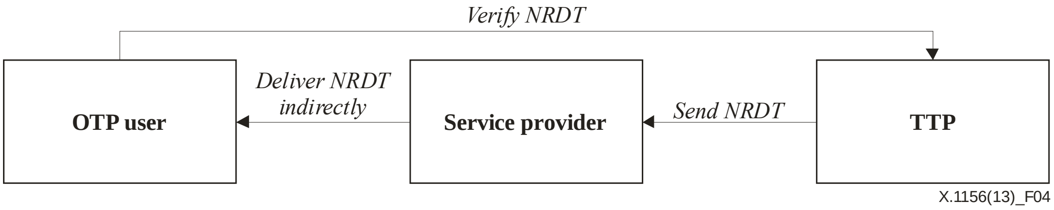

Figure 4 – Communication model for indirect NRDT

In most cases, the non-repudiation service needs both the NROT and the NRDT to maintain the fairness between the OTP user and the service provider. There are four possible non-repudiation service models consisting of direct NROT, direct NRDT, indirect NROT, and indirect NRDT. Service providers can selectively choose the suitable model according to their business model and the service environments.

10.2 Service model for indirect NROT and indirect NRDT

The OTP user requests an NROT from the TTP, and then the TTP generates the NROT and sends it to the OTP user. When the OTP user communicates with the service provider, the OTP user delivers the NROT to the service provider with the transaction data. Then the service provider verifies the NROT through the TTP. The service provider requests the NRDT from the TTP when the transaction data has been received from the OTP user; then the TTP generates the NRDT and sends it to the service provider. The service provider delivers the NRDT to the OTP user, and then the OTP user verifies the token through the TTP (see Figure 5).

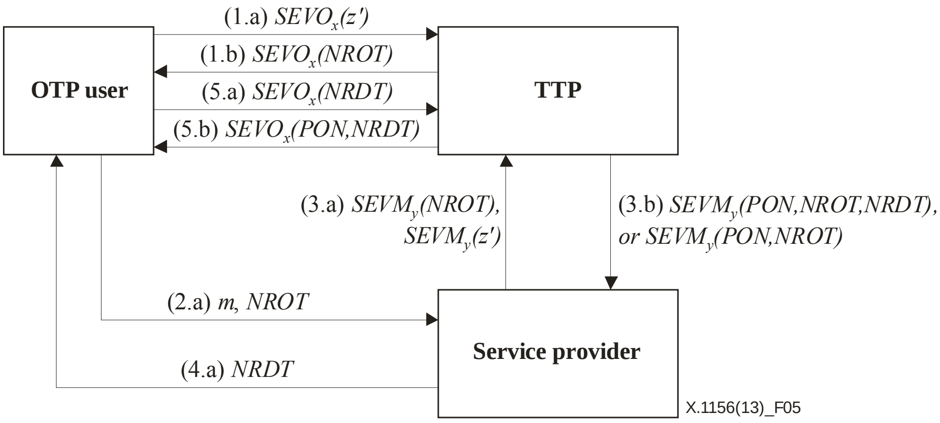

Figure 5 – Service model for indirect NROT and indirect NRDT

(1) Transaction 1 − Request NROT

- (1.a) The OTP user generates an OTP-based secure envelope $SEVO_X(z')$. The key X is an OTP user's key, and z' refers to the message excluding T_g from z. Then the OTP user sends the secure envelope to the TTP.
- (1.b) The TTP verifies the secure envelope $SEVO_X(z')$ received from the OTP user. If the secure envelope is valid, the TTP completes z by combining z' and T_g, and generates an OTP-based non-repudiation of origin token (NROT). The TTP generates the secure envelope $SEVO_X(NROT)$ and then sends the secure envelope to the OTP user.

(2) Transaction 2 − Request a non-repudiation service

- (2.a) The OTP user delivers the OTP-based non-repudiation of origin token (NROT) and transaction data to the service provider.

(3) Transaction 3 − Verify NROT and response NRDT

- (3.a) The service provider verifies the transaction data, and then generates a secure envelope $SEVM_Y(NROT)$ and $SEVM_Y(z')$. The key Y is a service provider's key. The service provider sends the secure envelopes to the TTP.
- (3.b) The TTP verifies the secure envelopes $SEVM_Y(NROT)$, $SEVM_Y(z')$ and the token NROT. If both are valid, the TTP completes z by combining z' and T_g, and generates an OTP-based non-repudiation of delivery token NRDT. And the TTP generates the secure envelope $SEVM_Y(PON,NROT,NRDT)$, where PON is positive that is the verification result. Then the TTP sends the secure envelope to the service provider. If the secure envelope $SEVM_Y(NROT)$ is valid but the token NROT is not, the TTP sends $SEVM_Y(PON,NROT)$ to the service provider, where PON is negative.

(4) Transaction 4 − Delivery token

- (4.a) The service provider delivers an NRDT to the OTP user.

(5) Transaction 5 − Verify NRDT

- (5.a) The OTP user generates the secure envelope $SEVO_X(NRDT)$ using the OTP user's key. The OTP user sends the secure envelope to the TTP.
- (5.b) The TTP verifies the secure envelope $SEVO_X(NRDT)$ and the NRDT received from the OTP user. If both are valid, the TTP generates the secure envelope $SEVO_X(PON,NRDT)$, where PON, the verification result, is positive. The TTP then sends the secure envelope to the OTP user. If the NRDT is not valid, the TTP sends the secure envelope $SEVO_X(PON,NRDT)$ where PON is negative. Finally, the OTP user saves the NRDT.

10.3 Service model for indirect NROT and direct NRDT

The OTP user requests an NROT from the TTP, and then the TTP generates an NROT and sends it to the OTP user. When the OTP user communicates with the service provider, the OTP user delivers the NROT to the service provider with the transaction data. Then the service provider verifies the NROT through the TTP. The service provider requests an NRDT from the TTP when the transaction data has been received from the OTP user. The TTP then generates an NRDT and sends it to the service provider and the OTP user. It is not necessary for the OTP user to verify the NRDT (see Figure 6).

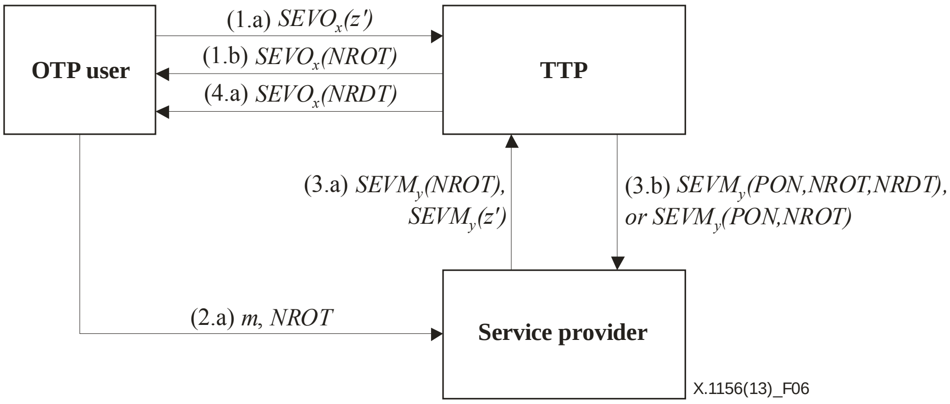

Figure 6 – Service model for indirect NROT and direct NRDT

(1) Transaction 1 – Request NROT

- (1.a) The OTP user generates an OTP-based secure envelope $SEVO_X(z')$. The key X is an OTP user's key, and z' refers to the message excluding T_g from z. Then the OTP user sends the secure envelope to the TTP.
- (1.b) The TTP verifies the secure envelope $SEVO_X(z')$ received from the OTP user. If the secure envelope is valid, the TTP completes z by combining z' and T_g, and generates an OTP-based non-repudiation of origin token (NROT). The TTP generates the secure envelope $SEVO_X(NROT)$ and then sends the secure envelope to the OTP user.

(2) Transaction 2 – Request a non-repudiation service

- (2.a) The OTP user delivers the OTP-based non-repudiation of origin token (NROT) and transaction data to the service provider.

(3) Transaction 3 – Verify NROT and response NRDT

- (3.a) The service provider verifies the transaction data, and then generates a secure envelope $SEVM_Y(NROT)$ and $SEVM_Y(z')$. The key Y is a service provider's key. The service provider sends the secure envelopes to the TTP.
- (3.b) The TTP verifies the secure envelope $SEVM_Y(NROT)$, $SEVM_Y(z')$ and the token NROT. If both are valid, the TTP completes z by combining z' and T_g, and generates an OTP-based non-repudiation token NRDT. The TTP generates the secure envelope $SEVM_Y(PON,NROT,NRDT)$, where PON, the verification result, is positive. Then the TTP sends the secure envelope to the service provider. If the secure envelope $SEVM_Y(NROT)$ is valid but the token NROT is not valid, the TTP sends $SEVM_Y(PON,NROT)$ to the service provider where PON is negative.

(4) Transaction 4 – Delivery token

- (4.a) The TTP sends $SEVO_X(NRDT)$ to the OTP user immediately after sending an NRDT to the service provider in transaction 3. The OTP user verifies $SEVO_X(NRDT)$. If the secure envelope is valid, the OTP user saves the NRDT.

10.4 Service model for direct NROT and indirect NRDT

The OTP user requests an NROT from the TTP, and then the TTP generates the NROT and directly sends it to the service provider. In this case, it is not necessary for the service provider to verify the NROT. The service provider requests an NRDT from the TTP where the transaction data is received from the OTP user; then the TTP generates an NRDT and sends it to the service provider. The service provider delivers an NRDT to the OTP user, and then the OTP user verifies the NRDT through the TTP (see Figure 7).

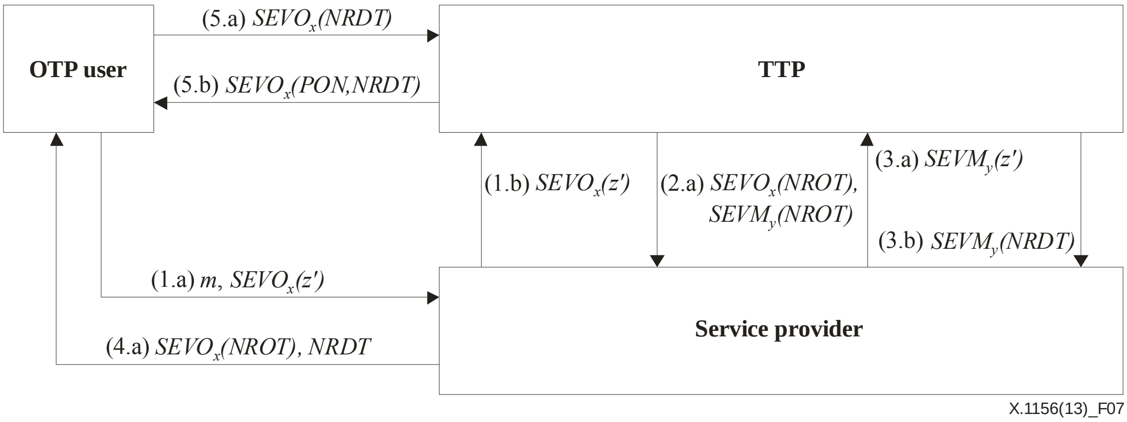

Figure 7 – Service model for direct NROT and indirect NRDT

(1) Transaction 1 – Request NROT

- (1.a) The OTP user generates an OTP-based secure envelope $SEVO_X(z')$. The key X is an OTP user's key, and z' refers to a message excluding T_g from z. Then the OTP user sends the OTP-based secure envelope $SEVO_X(z')$ and transaction data to the service provider.
- (1.b) The service provider verifies the transaction data. If the transaction data is valid, the service provider delivers the OTP-based secure envelope $SEVO_X(z')$ to the TTP. However, if the transaction data is not valid, the protocol stops immediately.

(2) Transaction 2 – Response NROT

- (2.a) The TTP verifies the OTP-based secure envelope $SEVO_X(z')$ received from the OTP user through the service provider. If the secure envelope is valid, the TTP completes z by combining z' and T_g, and generates an OTP-based non-repudiation of origin token (*NROT*). The TTP generates the secure envelope $SEVO_X(NROT)$ and $SEVM_Y(NROT)$ and then sends the secure envelope to the service provider.

(3) Transaction 3 – Verify NROT and response NRDT

- (3.a) The service provider generates a secure envelope $SEVM_Y(z')$ to request the token *NRDT*. The key Y is a service provider's key. The service provider sends the secure envelope to the TTP.
- (3.b) The TTP verifies the secure envelope $SEVM_Y(z')$. If the secure envelope is valid, the TTP completes z by combining z' and T_g, and generates an OTP-based non-repudiation token *NRDT*. The TTP generates the secure envelope $SEVM_Y(NRDT)$. Then the TTP sends the secure envelope to the service provider. If the secure envelope $SEVM_Y(z')$ is not valid, the protocol stops immediately.

(4) Transaction 4 – Token delivery

- (4.a) The service provider verifies the secure envelope $SEVM_Y(NRDT)$ delivered by the TTP. If the secure envelope is valid, the service provider sends $SEVO_X(NROT)$ and *NRDT* to the OTP user. If the secure envelope $SEVM_Y(NRDT)$ is not valid, the protocol stops immediately.

(5) Transaction 5 – Verify NRDT

- (5.a) The OTP user verifies the secure envelope *$SEVO_X$(NROT*) and saves *NRDT*. When the OTP user needs to verify the token NRDT, the OTP user can verify the token through the TTP. The OTP user generates secure envelope *$SEVO_X$(NRDT)* using the OTP user's key and sends the secure envelope to the TTP.
- (5.b) The TTP verifies the secure envelope *$SEVO_X$(NRDT)* and *the* NRDT received from the OTP user. If both are valid, the TTP generates the secure envelope *$SEVO_X$(PON,NRDT)*, where PON, the verification result is positive. The TTP sends the secure envelope to the OTP user. If the secure envelope *$SEVO_X$(NRDT)* and NRDT are not valid, the TTP sends the secure envelope *$SEVO_X$(PON,NRDT)* where PON is negative.

10.5 Service model for direct NROT and direct NRDT

The OTP user requests an NROT from the TTP; and then the TTP generates an NROT and directly sends it to the service provider. In this case, it is not necessary for the service provider to verify the NROT. The service provider requests an NRDT from the TTP where the transaction data is received from the OTP user; then the TTP generates an NRDT and sends it to the service provider and the OTP user. It is not necessary for the OTP user to verify the NRDT (see Figure 8).

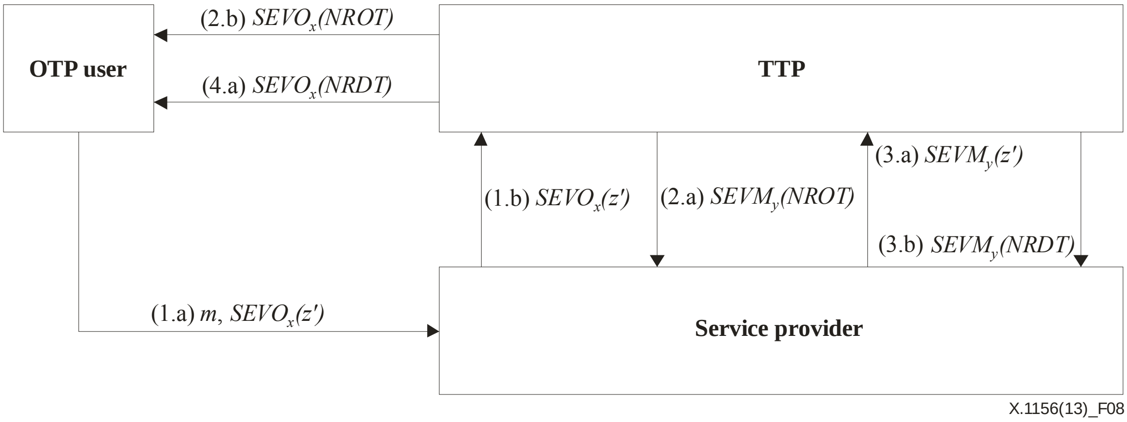

Figure 8 – Service model for direct NROT and direct NRDT

(1) Transaction 1 – Request NROT

- (1.a) The OTP user generates an OTP-based secure envelope *$SEVO_X$(z')*. The key X is an OTP user's key, and z' refers to a message excluding T_g from z. Then the OTP user sends the OTP-based secure envelope *$SEVO_X$(z')* and transaction data to the service provider.
- (1.b) The service provider verifies the transaction data. If the transaction data is valid, the service provider delivers the OTP-based secure envelope *$SEVO_X$(z')* to the TTP. However, if the transaction data is not valid, the protocol stops immediately.

(2) Transaction 2 – Response NROT

- (2.a) The TTP verifies the OTP-based secure envelope *$SEVO_X$(z')* received from the OTP user through the service provider. If the secure envelope is valid, the TTP completes z by combining z' and T_g, and generates an OTP-based non-repudiation of origin token (NROT). The TTP generates the secure envelope *$SEVO_X$(NROT)* and *$SEVM_Y$(NROT)* and then sends the secure envelope *$SEVM_Y$(NROT)* to the service provider.
- (2.b) The TTP sends *$SEVO_X$(NROT)* to the OTP user.

(3) Transaction 3 – Response NRDT

- (3.a) The service provider generates a secure envelope *$SEVM_Y(z')$* to request the token NRDT. The key Y is a service provider's key. The service provider sends the secure envelope to the TTP.
- (3.b) The TTP verifies the secure envelope *$SEVM_Y(z')$*. If the secure envelope is valid, the TTP completes z by combining z' and T_g, and generates an OTP-based non-repudiation token NRDT. The TTP generates the secure envelope *$SEVM_Y(NRDT)$*. Then the TTP sends the secure envelope to the service provider. If the secure envelope *$SEVM_Y(z')$* is not valid, the protocol stops immediately.

(4) Transaction 4 – Token delivery

- (4.a) The TTP generates the secure envelope *$SEVO_X(NRDT)$* and then sends the secure envelope to the OTP user. The OTP user verifies the secure envelope. If the secure envelope is valid, the OTP user saves NRDT.

Bibliography

[b-ITU-T X.810] Recommendation ITU-T X.810 (1995) | ISO/IEC 10181-1:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Overview*.

[b-ITU-T X.811] Recommendation ITU-T X.811 (1995) | ISO/IEC 10181-2:1996, *Information technology – Open Systems Interconnection − Security frameworks for open systems: Authentication framework*.

[b-ITU-T X.842] Recommendation ITU-T X.842 (2000) | ISO/IEC TR 14516:2002, *Information technology – Security techniques – Guidelines for the use and management of trusted third party services*.

[b-ITU-T X.1141] Recommendation ITU-T X.1141 (2006), *Security Assertion Markup Language (SAML 2.0)*.

[b-ISO/IEC 13888-3] ISO/IEC 13888-3:2009, *Information technology – Security techniques – Non-repudiation – Part 3: Mechanisms using asymmetric techniques*.

[b-IETF RFC 2289] IETF RFC 2289 (1998), *A One-Time Password System*.

[b-IETF RFC 4226] IETF RFC 4226 (2005), *HOTP: An HMAC-Based One-Time Password Algorithm*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2013