भारतीय मानक

Indian Standard

IS 18900 (Part 2) : 2024
ISO 19626-2 : 2021

प्रशासन और वाणिज्य उद्योग में डाटा तत्वों और दस्तावेजों की कार्यविधि — इलेक्ट्रॉनिक दस्तावेजों के लिए विश्वसनीय संचार प्लेटफार्म

भाग 2 अनुप्रयोग

Processes Data Elements and Documents in Commerce Industry and Administration — Trusted Communication Platform for Electronic Documents

Part 2 Applications

ICS 35.240.63

भारतीय मानक ब्यूरो
BUREAU OF INDIAN STANDARDS
मानक भवन, 9 बहादुर शाह ज़फर मार्ग, नई दिल्ली - 110002
MANAK BHAVAN, 9 BAHADUR SHAH ZAFAR MARG
NEW DELHI - 110002
www.bis.gov.in www.standardsbis.in

November 2024 **Price Group 15**

Documentation and Information Sectional Committee, MSD 05

NATIONAL FOREWORD

This Indian Standard which is identical to ISO 19626-2 : 2021 ‘Processes, data elements and documents in commerce, industry and administration — Trusted communication platform for electronic documents — Part 2: Applications’ issued by the International Organization for Standardization (ISO) was adopted by the Bureau of Indian Standards on the recommendation of the Documentation and Information Sectional Committee and approval of the Management and Systems Division Council.

The text of the ISO standard has been approved as suitable for publication as an Indian Standard without deviations. Certain conventions are, however, not identical to those used in Indian Standards. Attention is particularly drawn to the following:

a) Wherever the words ‘International Standard’ appear referring to this standard, they should be read as ‘Indian Standard’; and

b) Comma (,) has been used as a decimal marker while in Indian Standards, the current practice is to use a point (.) as the decimal marker.

The Committee has reviewed the provisions of the following International Standard referred in this adopted standard and has decided that it is acceptable for use in conjunction with this standard:

International Standard	*Title*
ISO 19626-1	Processes, data elements and documents in commerce, industry and administration — Trusted communication platforms for electronic documents — Part 1: Fundamentals

Annex A, Annex B and Annex C are for information only.

In reporting the results of a test or analysis made in accordance with this standard, if the final value, observed or calculated, is to be rounded off, it shall be done in accordance with IS 2 : 2022 ‘Rules for rounding off numerical values (*second revision*)’.

Contents

Page

Introduction

This document presents the TCP (trusted communication platform) system for trusted communication in the open and distributed ICT (information communication technology) environment, as a connected standard of ISO 19626-1.

The TCP system is a kind of middleware for connecting trusted communication in IoT (internet of things) or cloud environments, that delivers the information between humans, organizations, and devices by exchanging the e-documents via the TCP system components and stores the evidence of executed communication.

This document specifies the functionalities of processes and APIs (application programming interfaces) between TCP system components.

It intends to be described in the technology-neutral way in order that a TCP system can be implemented by applying various wire-wireless applied services and communication protocols used in the real world.

The key points that are implicated to this document are as follows.

a) The communication protocol used for inter-connection between TCP components is a core function of the application service layer in the distributed environment of wire and wireless communication.

The basic function of sending or receiving messages between the TCP system components compose the common communication interface to deliver message(s) in a distributed computing system of wire and wireless environment.

b) TCE (trusted communication evidence) can prove trusted communication in a TCP.

The TCP communication server executes reliable communication transactions, and create and store TCE as the proof in a way of non-repudiation between the communication participants.

c) A TCP system can be adequately ported to various kinds of business communication systems.

A TCP system is connected as a transmit or receive module between the e-business systems connected to be distributed with various work systems of B2B, e-government, and e-trade as well as the simple electronic communication systems to transmit contents directly using the address of sender or receiver (URLs, IP, address) such as the e-mail system as a related application system.

Indian Standard

PROCESSES DATA ELEMENTS AND DOCUMENTS IN COMMERCE INDUSTRY AND ADMINISTRATION — TRUSTED COMMUNICATION PLATFORM FOR ELECTRONIC DOCUMENTS

PART 2 APPLICATIONS

1 Scope

As a connected standard of ISO 19626-1, this document defines the communication interactions between TCP system components and specifies their detailed interfaces — the processes and the APIs of the TCP system components.

It provides the common communication interface for deployment and implementation of the system components, and their functions in a specific technology-neutral way to those who consider applying and establishing a TCP system.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 19626-1, *Processes, data elements and documents in commerce, industry and administration — Trusted communication platforms for electronic documents — Part 1: Fundamentals*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 19626-1 and the following apply.

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

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

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

3.1
blacklist
list of *e-identities* (3.3) of the originators who are proved having 'malicious intent'

Note 1 to entry: If a message is confirmed as *spam* (3.5), an e-identity who sent the spam is classified as a sender having 'malicious intent'.

Note 2 to entry: An addressee receiving a message from the originator in the blacklist can reject receiving the message.

3.2
characteristic information
unique identifying information to identify the entity in the offline (real) world such as a resident registration number, social security number, or identification number of an IoT device

3.3
e-identity
sole object to identify the entity who is the actual subject of communication activity under a TCP system

Note 1 to entry: In a TCP, it is the object which expresses the entity who is the actual subject of all activities including transmission, reception, and perusal (viewing or reading), etc. of e-documents after the electronic verification of identity.

3.4
e-identity ID
name that refers to an *e-identity* (3.3) identifying a value an e-identity gives itself for identification

Note 1 to entry: With the ID, the e-identity expresses itself and distinguishes itself from other e-identities.

3.5
spam
unsolicited email, which can carry malicious contents and/or scam messages

[SOURCE: ISO/IEC 27033-1:2015, 3.37, modified — "unsolicited emails" has been replaced with "unsolicited email".]

3.6
whitelist
list of trusted communication servers in a TCP

Note 1 to entry: If a communication server is proved that the one is secure technically and politically and complies with a standard and policy of the TCP, then TTP (trusted third party) directory server adds the one to its whitelist.

4 Relational architecture of TCP

4.1 Overview

ISO 19626-1 presents 2 types of 'TCP main' and 'TCP client' in system architecture. As a connected standard, this document enhances its relational architecture at the view of the interface.

As shown in Figure 1, once a transmitting entity (i.e. a sender) makes a delivery request to a receiving entity (i.e. a receiver), each of the components can be linked to one another through linkage interfaces, and the communication server is enabled to form an entrusted chain with a relying party. The pair-linked communication servers implement communication that can be entrusted, and through their interactions, generate TCE and can possess evidence in the TCE repository.

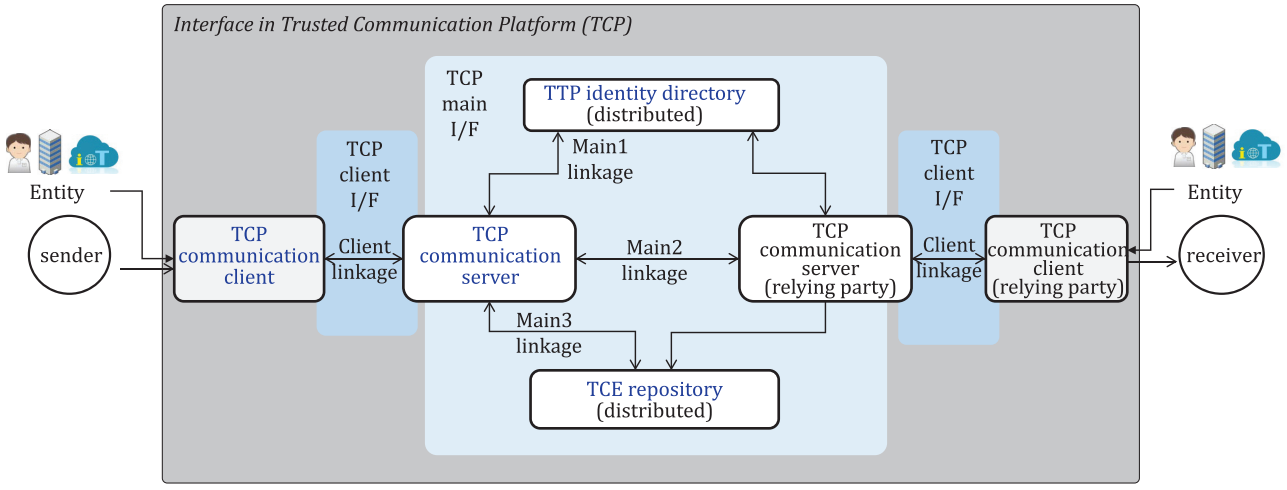


Figure 1 — TCP relational architecture

4.2 TCP relational architecture

Even if some communities intend to establish a TCP, they could not implement it in case their business and technical environments are different.

The particular authentication level, applied technology and communication protocol, etc. in each linkage need to be arranged properly by designing after classifying 'TCP main' and 'TCP client' even under various existing legacy system environments (refer to the ISO 19626-1:2020, 5.2). Figure 1 shows two interfaces.

a) TCP client interface

'TCP client interface' refers to an area inter-linked between a TCP communication client and a TCP communication server. Apart from various existing legacy system environments, a TCP communication client chooses and delegates a TCP communication server as its agent for trusted communication. At this point, 'TCP client interface' should be agreed and linked by the SLA (service level agreement) suggested by the communication server. Thus, in this interface, the communication server can function an agent of the communication client to transmit the requested e-document(s) in a trusted manner under a TCP architecture.

A TCP requires a standard interface for common linkage that a communication client and a server shall comply with. Then there are advantages of being able to provide convenience or efficiency of TCP operation to the communication clients. If a communication client wants to change its agent into the other communication server in a TCP, the communication client is able to change easily with it without being dependent on the proprietary interface of a specific communication server.

1) Client linkage: between a TCP communication client and a TCP communication server

— Once the entity gets to register its own e-identity by going through the process of verifying it from the TTP identity directory, this entity becomes a participant as a TCP communication client.

— A TCP communication client can participate in trusted communication after signing a service agreement provided by the communication server. This means the communication client does not perform the direct communication with the other communication client(s).

— A TCP communication client can delegate trusted communication after authentication of the TCP communication server in the PR3 (communication authentication process).

b) TCP main interface

'TCP main interface' refers to an area which performs practical trusted communication through three linkages that shall comply with a communication interface specification (see Reference [6]). 'TCP main' has the following types of linkage:

1) Main1 linkage: between a TCP communication server and a TTP identity directory

— For the communication server to send or receive e-documents on the behalf of communication client, information on the TCP communication server shall be registered in the TTP identity directory in the PR1 (communication server registration process).

— The newly registered TCP communication server shall get added to the whitelist as a trusted list in the identity directory. Then the identity directory shall notify the changed whitelist to the other registered communication servers in the PR2 (e-identity registration process).

— Communication server shall query to the identity directory in order to acquire and verify information on the relying party of reception in the PR4 (e-document transmitting process).

2) Main2 linkage: between TCP communication servers

— When a communication server transmits e-documents by inter-linking with the communication server of relying party, this communication server acts as a transmitting server.

— When a communication server has received the e-document, this communication server acts as a receiving server by processing it in the PR4 (e-document transmission process).

3) Main3 linkage: between a TCP communication server and a TCE repository

— Communication server(s) shall store the TCE generated after sending or receiving an e-document as evidence on the transactions of sending or receiving in the TCE repository in the PR6 (TCE preservation process).

— If verification on the communication of sending or receiving the e-document is necessary, TCE repository can verify the communication based on the stored TCE in the process of communication server verification.

4.3 Functionalities of TCP components

4.3.1 TTP identity directory

4.3.1.1 General

TTP identity directory provides a service to store and retrieve e-identity information on the entity after identifying and authenticating the entity participating in trusted communication in a reliable method. The entity becomes a member of TCP as a communication client after registering an e-identity in the TTP identity directory. In one TCP, only one TTP identity directory that has e-identity information on all communication clients shall exist logically. In other words, even if the e-identity information is physically distributed or replicated information exists in various places, there should be only one integrated e-identity information logically and one shall be able to obtain the same information no matter when or by whom the information is searched or retrieved.

TTP identity directory provides the 5 functions defined in 4.3.1.2 to 4.3.1.6.

4.3.1.2 To register and manage trusted list of TCP communication server

— A TCP communication server shall perform the function to transmit or receive e-documents by receiving the request of the communication client. For doing it, this server shall be registered in the TTP identity directory.

— Before the TTP identity directory registers a TCP communication server, methods or procedures to verify functional security requirements, conformity of standards and interoperability shall be determined according to 'TCP main' policy. However, such a policy of the TTP identity directory shall reach a mutual agreement between the participants of TCP.

— After the communication server goes through verification on whether the concerned server is implemented by conforming to the standard and whether the necessary functional requirements are implemented, the network address of communication server and the information necessary for security, etc. shall be registered at the trusted list in the TTP identity directory.

— The trusted list of registered communication servers is managed as the whitelist and only the communication server listed in the whitelist can participate in trusted communication. The whitelist consists of a trusted list of TCP communication servers in the process of communication server registration.

4.3.1.3 To identify entity

— TTP identity directory shall check and authenticate whether the information provided by the entity is identical to its actual information in the real world (e.g. if the entity is a person or an organization, name or unique ID of the entity such as resident registration number, social security number or DUNS number, etc. and in case of a IoT device, it includes device ID, IP number and etc.) in the process of registering, modifying or deleting e-identity information.

— Criteria or methods for verifying the identity of an entity are determined according to the policy of the TTP identity directory and these shall be agreed between the participants who are performing trusted communication under the concerned TCP system.

4.3.1.4 To register and manage information of entity

— To perform trusted communication under a TCP system, the entity shall register e-identity information to the TTP identity directory.

— The entity may be a person or a conceptual subject such as a company, an organization, or IoT device, etc.

— For the entity to register its information, information on which communication server is used for sending or receiving e-documents in trusted mode is also necessary in addition to the basic information on the entity such as unique ID which represents an e-identity, entity name, and an ID commonly used in the real world (offline).

— In TCP, an entity is represented as an e-identity; and only an entity that has registered its e-identity may participate in trusted communication of e-documents as a TCP communication client.

4.3.1.5 To search e-identity information

— If the transmitting client intends to send an e-document to a receiving client in TCP, the transmitting server which receives a request of sending an e-document from the transmitting client shall query to the TTP identity directory in order to obtain information on the receiving server which receives e-documents on the behalf of the receiving client.

— For this, the transmitting server requests to retrieve information which includes the network address of the receiving server used by the receiving client to the TTP identity directory using the e-identity ID value of the receiving client. After retrieving the requested information, the TTP identity directory returns the retrieved information to the transmitting server.

— Also, in order to verify whether the transmitting server that has sent the message is the legitimate communication server performing the role as an agent of transmission for the transmitting client at the time of receiving the message, the receiving server shall query on this to the TTP identity directory.

4.3.1.6 To handle spam messages, blacklist and whitelist

— Once the received message is determined as a spam message, the receiving client reports this message as a spam message to the TTP identity directory through the receiving server. The identity directory shall review the spam message status of this message after receiving the report of the spam message.

— Once the TTP identity directory determines the reported message as the spam message, the TTP identity directory shall add the originator (i.e. the e-identity of transmitting client) of the concerned message in blacklist and shall notify the updated blacklist to all communication servers in TCP. Unlike the whitelist managed as a list of communication servers, the blacklist is registered and managed as a list of e-identities.

— Criteria or procedures to decide whether the submitted report of the spam message is appropriate are determined according to the policy of the TTP identity directory and shall be agreed between TCPSPs (TCP service providers) who are performing trusted communication under the concerned TCP system.

4.3.2 TCP communication server

4.3.2.1 General

TCP communication server provides a service to send or receive e-documents using a trusted method by receiving a request of communication clients under a TCP system. All communication servers in one TCP shall be implemented according to mutually agreed transmission or reception protocols inside the TCP. Accordingly, all communication servers shall be verified in advance on whether the system

operates by conforming to the standards agreed in TCP main and whether it is interoperable with other components in order to participate in TCP.

Methods or procedures to verify conformity with standards or interoperability on the communication server shall be determined by mutual agreement between the TCPSPs.

TCP communication server shall provide the functions defined in 4.3.2.2 to 4.3.2.11.

4.3.2.2 To register and manage TCP communication client

— TCP communication client shall sign on an agreement about the use of trusted transmission or reception service of e-documents provided by the TCP communication server to delegate actions of trusted communication to the communication server.

— For doing this, the communication server shall provide a function for the communication client to apply for the use of services and a function to manage the information of communication clients with whom the communication server makes an agreement on the use of services.

— For the communication client to apply for the use of services to the communication server, a client shall be registered as an e-identity to the TTP identity directory and shall present a unique ID (i.e. e-identity ID) representing the e-identity registered to the identity directory when applying for the use of services.

— The communication server shall go through the process of verifying whether the connecting communication client currently is a legitimate owner of the e-identity ID presented by a communication client when applying for the use of services.

— After being registered to the TCP communication server properly, a TCP communication client will be able to use the trusted transmission or reception service of e-documents provided by the communication server.

4.3.2.3 To authenticate TCP communication client for requesting the services

— An authentication process on the communication client is absolutely necessary, so that communication server acts as an agent on the service in TCP by receiving a request from the communication client. In other words, the communication server shall know which e-identity the communication client requests the services with.

— To verify whether the client requesting usage of the services to the communication server is registered or not, the communication server shall perform authentication using various methods such as performing authentication using ID/PW, personal information or biometrics information of client.

— If a TCP client is authenticated successfully, the communication server, as an agent of communication client, performs the services related to trusted communication of e-documents such as transmission, reception, perusal or a spam message report.

4.3.2.4 To create trusted chain for TCP communications

— Once the transmitting client requests transmission of messages to the transmitting server, the transmitting server shall authenticate the e-identity of the transmitting client first to prevent from deceiving the receiving client as if the transmitting client is another user.

— As the next step, the transmitting server requests the network address (such as IP address) of the receiving server to receive messages on behalf of the receiving client to the TTP identity directory using the e-identity ID of the receiving client presented by the transmitting client.

— The receiving server shall verify whether the transmitting server that has transmitted messages is the server properly registered in the whitelist of the TTP identity directory under TCP system.

4.3.2.5 To transmit messages

— The TCP communication server shall transmit the document by the request of the transmitting client to the receiving client using a trusted method.

— For doing so, the communication server shall comply with all security and reliability requirements of the transmission process including proper verification of identity on the transmitting client, packaging of a trusted method on the delivered document, reliability on the identity information of the transmitting client, securing the integrity of transmitted messages, guaranteeing confidentiality in the process of trusted communication between the transmitting client or receiving client, and even the verification on whether the receiving client has received the e-document sent by the transmitting client.

NOTE Integrity of transmitted message means to verify that the document transmitted is safely delivered to the receiving client from the transmitting client without being forged in the process of delivering document.

4.3.2.6 To receive messages

— The communication server as a receiving server shall respond after verifying whether the-transmitted message from the transmitting server is a trusted message created according to the TCP transmission protocol.

— To make this possible, the receiving server shall verify the reliability on the transmitting client information in the received message, the integrity of the received message, confidentiality in the process of communication between the transmitting client and receiving client. etc. After verifying that the received message is trusted, it should secure justifiability and reliability on the reception of the message through the ACK (acknowledgment) for receipt confirmation.

4.3.2.7 To store and manage transmitted and received messages

— The communication server can safely store all messages transmitted or received under the TCP system by the request of the communication client.

— The communication server can set up the period to store transmitted or received messages and the scope of communication clients to access (search, browse or delete) the stored message according to the policy agreed at the time of concluding an agreement with the communication client.

— The communication server shall be managed to avoid the stored messages from getting leaked or damaged wrongfully by another communication client which is not the communication client for whom the access is permitted by the agreement with the communication client.

4.3.2.8 To handle spam message reports

— If the communication client requests to the communication server to report a specific received message as a spam message, the communication server reports the message as a spam to the TTP identity directory.

— Once the communication server receives the examination result on the reported message as a spam from the TTP identity directory, the communication server shall notify this result to the communication client that had requested the report of the spam.

4.3.2.9 To create and deliver the NRR (non-repudiation of receipt)/NRD (non-repudiation of delivery) for receipt confirmation

— The receiving server shall create the NRR/NRD including the ACK signal to confirm that the receiving server has received the message at the time of receiving the message; and the NRR consists of the information of transmitting client, information of receiving client, transmitted date/time, received date and the information to prove that the contents of e-document is not forged (e.g. the hash value of e-document). NRD consists of the information of perusal confirmation (e.g. perused date/time, the hash value of perused e-document).

— In order that the transmitting server is able to get evidence that the e-document sent by the transmitting client is delivered properly to the receiving client, the receiving server shall deliver the NRR/NRD to the transmitting server instantly after creating the NRR/NRD.

4.3.2.10 To receive NRR/NRD and create TCE

— The transmitting server shall receive the NRR/NRD created by the receiving server after transmitting the e-document to complete the transmitting process successfully.

— The transmitting server shall verify whether the details on receipt confirmation included in the NRR/NRD is accurate, and whether this has been confirmed by the receiving server, based on the message of transmitting the e-document earlier.

— If the received NRR/NRD is valid, the transmitting server shall create TCE including this NRR/NRD.

4.3.2.11 To request storage and verification of TCE

— After creating the TCE including the NRR/NRD for receipt and/or perusal confirmation, the communication server participated in trusted communication requests to store this into the TCE repository.

— The participants who have participated in communication or a third party that needs evidence shall request verification on the fact of sending or receiving e-document based on TCE stored in the TCE repository.

4.3.3 TCP communication client

4.3.3.1 General

'TCP communication client' means a component that performs the role of the entity's proxy under the TCP system for the entity to use the services of communication server to transmit or receive e-documents using the trusted method. Since a communication client may not transmit or receive e-documents, it shall use the concerned service after making sure to sign an agreement on the use of e-document transmission or reception service with TCP communication server in order to participate in trusted communication.

TCP Communication client should provide the functions defined in 4.3.3.2 to 4.3.3.6.

4.3.3.2 To request authentication for using TCP communication services

— The communication client shall provide the function to deliver information for authentication to the communication server in order to get the approval of using the communication services provided by the communication server. For this, the communication client shall be registered to the communication server ahead.

— The communication client shall be allowed to use the functions such as requests to transmit or receive e-documents only in case of the communication client authenticated by the communication server.

4.3.3.3 To request transmission

— The transmitting client shall provide the function to make the transmission request information for transmitting e-documents to the receiving client.

— The transmitting client shall request the transmission of e-document by connecting to the transmitting server.

— The transmitting client shall provide the function to receive information of transmission status from the transmitting server after requesting the transmission of e-documents.

4.3.3.4 To get the received e-documents

— The receiving client shall provide the function to get the list of received messages, the detail information of specific message and the attached e-document by retrieving them from receiving server.

4.3.3.5 To request communication verification based on TCE

— When an entity desires to receive evidence to prove that the information of communication on the transmitted or received message is true, the communication client shall provide the function to deliver such a request to the communication server.

— The communication client that has received a request for communication verification from the entity shall request to verify the communication with the information of the communication (such as message ID) to the communication server and return the verification result received from TCE repository through the communication server.

4.3.3.6 To report spam messages

— The communication client shall provide a function to report spam messages if it (especially sending client) determines the received message as a spam message after perusing it.

— If a communication client (especially sending client) selects the received message and intents to report this as a spam message, it shall request to report the message as the spam message to the TTP identity directory through the communication server.

— The communication client (especially sending client) shall inquire the examination result of the spam status on the message which the communication client has reported as the spam message through the communication server.

4.3.4 TCE repository

4.3.4.1 General

'TCE repository' stores and manages TCE which proves the fact of (communication) transmitting or receiving messages between communication servers. It plays a role of verifying the fact of communication and assigning trust on the verified information based on TCE when requested by someone else. TCE repository may be operated by one trusted third party or may be operated by getting distributed to TCPSPs. For example, most communication server might share TCE in each TCE repository such as blockchain. However, TCE repository shall have a precondition on the fact of being able to trust each other between all participants of TCP on the fact that the stored TCE is being safely managed from the threat of any forgery.

TCE repository should provide the functions defined in 4.3.4.2 to 4.3.2.3.

4.3.4.2 Storage and management of TCE

— After receiving and verifying the ACK signal including confirmation about information of communication obtained from the receiving server and creating TCE, the transmitting server shall request to store into the TCE repository. The TCE repository shall store TCE requested to be stored using a method which is safe from forgery.

— The period for the TCE repository to store TCE shall be determined by mutual agreement between the participants of TCP.

4.3.4.3 Verification on communication record

— If someone (e.g. communication server, 3rd party) requests verification of the communication record that had transmitted and received the e-document under TCP system (transmitting client information, receiving client information, transmitted date, received date and conformity of

transmitted or received document contents, etc.), the communication server shall provide the function to verify validity on the concerned information based on the stored TCE.

— If someone (e.g. communication server, 3rd party) requests evidence about communication record, the TCE repository shall provide a material consisting of the confirmation that 'verification on whether TCE is created through certainty, completeness, and confidentiality of communication (refer to the ISO 19626-1:2020, 4.2) and the communication record has been valid'.

5 TCP processes

5.1 Overview of main processes

For trusted communication, every communication shall be performed according to the procedure and the order determined between each component of TCP. In each stage, there are processes that shall be performed mandatorily such as e-identity creation or e-document transmitting and the processes where the status of performance is determined by the choice of an entity such as perusal confirmation process.

Figure 2 shows the processes (PRs) that are basically necessary in order to accomplish trusted communication between TCP clients under a TCP.

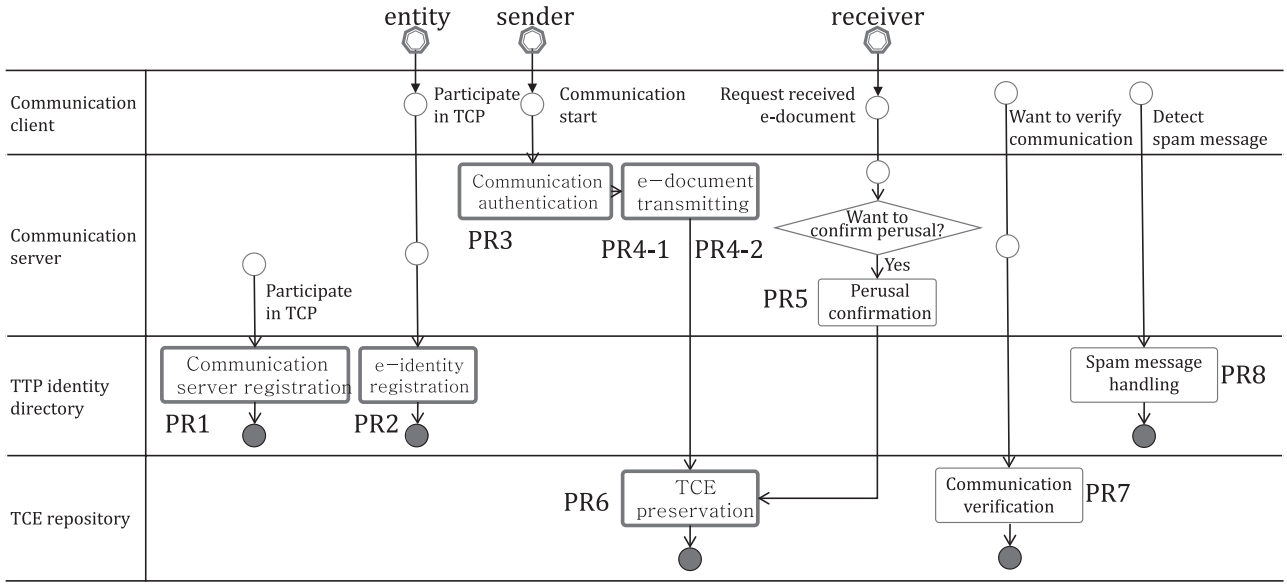


Figure 2 — Overview of TCP processes

Figure 2 emphasizes the necessary basic processes required for trusted communication. Of course, aside from these processes, various processes exist including the revision of information and inquiry of information. Each of the components within the TCP can provide the other processes (related to the myriad additional services as listed in the service agreement among the TCP participants) aside from processes defined in Figure 2; it can provide them with the resulting functions and APIs.

This document mainly describes the process that forms the core in the TCP for trusted communication. The processes are composed of 5 mandatory processes and 3 optional processes. The mandatory processes absolutely necessary for trusted communication are PR1 (communication server registration process), PR2 (e-identity registration process), PR3 (communication authentication process), PR4

(e-document transmitting process) and PR6 (TCE preservation process) and the processes generated optionally by an entity's request are PR5 (perusal confirmation process), PR7 (communication verification process) and PR8 (spam message handling process). Among these, PR4 and PR5 that are the most basic processes of trusted communication become one trusted communication unit inside TCP and grouped by a unique communication ID.

The purposes of the process(es) of each stage implemented are as follows:

a) PR1 (communication server registration process): In order to communicate using e-documents in a trusted manner in a TCP, all TCP communication servers to perform the roles of transmitting and receiving messages according to the standards and policies of TCP shall exist together; and such TCP communication servers shall be registered to the TTP identity directory as the trusted list.

b) PR2 (e-identity registration process): In order to have trusted communication under a TCP, both the entity trying to transmit the e-document and the entity receiving it shall have each unique identity (i.e. e-identity) before doing anything else; and they shall be registered to the TTP identity directory. After the entity completes registration with an e-identity, a service agreement for delegating trusted communication shall be signed related to trusted communication with the TCP communication server(s) providing the concerned services in order to transmit or receive e-documents.

c) PR3 (communication authentication process): A TCP communication client shall provide the communication server with information including an e-identity ID. It shall be authenticated on which entity is connecting to the communication server for using the service.

d) PR4 (e-document transmitting process): After the communication client that has obtained authentication from the communication server, the communication server shall transmit messages to the receiving client and confirm its successful transmission using a trusted method.

e) PR5 (perusal confirmation process): If the transmitting client may request to receive confirmation on whether the receiving client has perused the document, the information that confirms the perusal action of the receiving client is delivered to the transmitting client when the receiving client peruses the document. This process is an optional one which takes place only if the transmitting client desires to receive the perusal confirmation.

f) PR6 (TCE preservation process): To demonstrate the business transactions taken via PR4 (e-document transmitting process) or PR5 (perusal confirmation process), TCE is generated and then stored in the TCE repository.

g) PR7 (communication verification process): After performing trusted communication under a TCP, the information value of TCE can be validated before being stored in the TCE repository. This process is an optional one which takes place only if the TCE repository has this policy.

h) PR8 (spam message handling process): It's the process to prevent spreading spams in the TCP. This process is composed of the stage to report a spam message, the stage to determine whether the message is a spam by examining the reported spam message and the stage of prohibiting all messages transmitted by an originator of the spam message.

5.2 Description of each process

5.2.1 PR1 (communication server registration process)

For the communication server to perform trusted communication by acting as an agent of the communication client, it shall be registered to the TTP identity directory. Also, the communication server shall get a unique ID assigned according to the ID rule of the 'TCP main' when getting registered to the identity directory.

All communication servers registered in the TTP identity directory shall be managed as the whitelist and a communication server that couldn't be registered to the whitelist could not participate in trusted communication. For doing this, the TTP identity directory shall share information by distributing the

updated whitelist to all communication servers inside the TCP after reflecting on the whitelist if a new communication server is registered, or if information of a registered communication server is changed or deleted.

The overall feature of PR1 is shown in Figure 3.

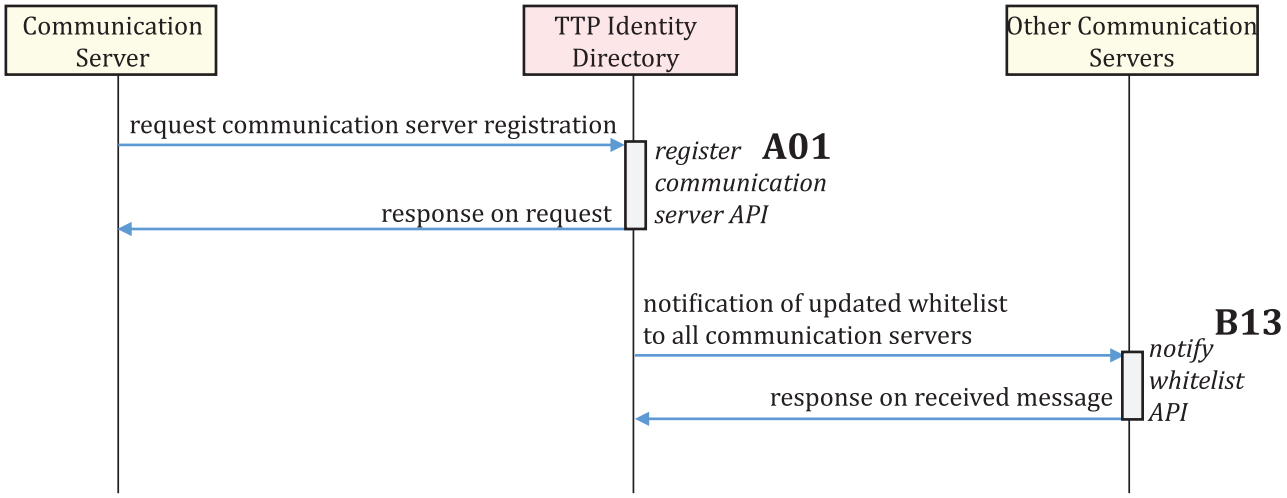


Figure 3 — PR1 (communication server registration process)

a) Related APIs

This process consists of A01 'register communication server' API provided by the TPP identity directory and B13 'notify whitelist' API provided by the communication server.

b) Flow of process

1) The communication server requests the TTP identity directory to register the communication server (call A01 'register communication server' API).

2) Once registration has been completed for the communication server that was requested to be registered, the TTP identity directory then reflects the registered information of the new communication server. After this, TTP identity directory shall notify the revised whitelist to another communication server (call B13 'notify whitelist' API).

3) The communication server that has received the notification for the revised whitelist then decides the target server for trusted communication based on this whitelist. In other words, under the TCP system, for trusted communication, all communication servers shall be included in the whitelist.

5.2.2 PR2 (e-identity registration process)

For the entity to participate in trusted communication, the entity shall create and register the e-identity to the TTP identity directory through a communication client. Now a communication client is a proxy of the entity in TCP. This process is the one to create and register e-identity for the entity to participate in the TCP.

After creating an e-identity, the communication client can use the following methods:

— requesting registration by connecting to the TTP identity directory directly; in this case, after registration of the e-identity, the entity shall sign on a service agreement with the communication server with the information of the registered e-identity before using the service; or

— requesting registration to the communication server after signing on a service agreement with the communication server; in this case, after receiving the request from the entity, the communication server shall substitute for the entity to request registration to the TTP identity directory.

The e-identity registration process is accomplished by two transactions called A04 'identify entity' API and A05 'register e-identity' API. If a communication client wants to create its e-identity before signing an agreement on the use of the TCP service between the entity and the server, then the subject of requesting registration of the e-identity may become the entity. But if an entity wants to create an e-identity after the signing on a service agreement, then the communication server may become a substitute for the entity to request the registration of the e-identity.

In order to identify the entity, the TTP identity directory shall request to the communication client the information to identify who the entity making registration request is in the real world and authenticate whether the entity is the legitimate owner of the information. Identification on this shall be determined by TCP main agreement between the members participating in the TCP and there may be various methods depending on the country or the local community.

NOTE For the identification method, identity of the entity can be verified using various appropriate methods, such as personal identification in cellular phone, public certificate, face-to-face authentication or identification through the information which only you know.

Figure 4 shows the overall feature of PR2.

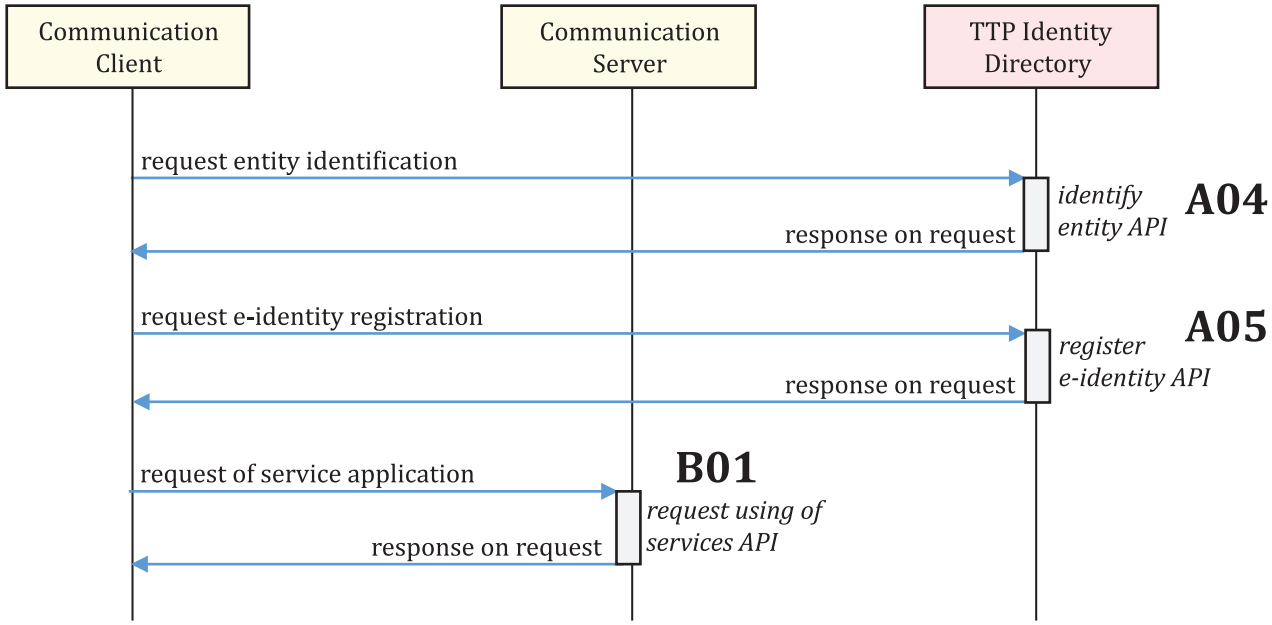


Figure 4 — PR2 (e-identity registration process)

a) Related APIs

This process consists of B01 'request using of services' API provided by the communication server, A04' identify entity' API and A05 'register e-identity' API provided by the TTP identity directory.

b) Flow of process

1) As an entity called an identifier under the TCP system, to generate a unique e-identity, a process first takes place to ensure that the entity connecting to the TTP identity directory through a communication client is the entity himself/herself/itself that exists in the real world. To do this, the communication client requests entity identification from the TTP identity directory (call A04 'identity entity' API).

2) Once identification for the entity has been completed, then the communication client as a proxy of the entity requests the TTP identity directory to register an e-identity after generating a unique e-identity (call A05 'register e-identity' API).

3) Once the e-identity has successfully been registered to the TPP identity directory, under the TCP system, instead of itself, the communication client will find a communication server for trusted communication of e-documents, and request the use of the services provided by the relevant server (call B01 'request using of services' API).

5.2.3 PR3 (communication authentication process)

For the communication client to perform the processes related to the transmission of e-documents, the perusal of received documents, or completeness of trusted communication, a communication client as a legitimate proxy of entity shall be authenticated by a communication server.

This process is the one to authenticate whether a communication client is a legitimate entity for authenticating the services of a communication server. If the communication client can't be authenticated as a legitimate proxy of an entity by the communication server, communication client may not request any functions related to trusted communication of e-documents to the communication server.

The communication authentication process is implemented according to the service authentication agreed upon by the communication client and communication server under PR2 (e-identity registration process). Also, after the communication authentication, under the TCP system, the communication client will act as the legitimate proxy for the e-identity of the authenticated entity.

Figure 5 shows the overall feature of PR3.

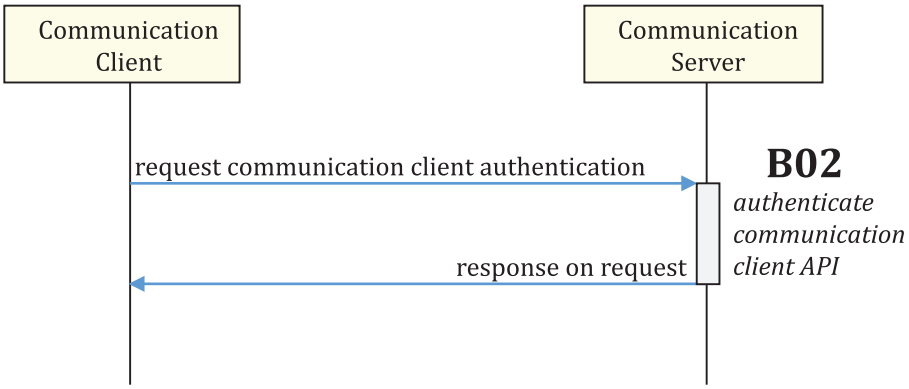


Figure 5 — PR3 (communication authentication process)

a) Related APIs

This process consists of B02 'authenticate communication client' API provided by the communication server.

b) Flow of process

1) To obtain authentication for the use of the service from the communication server, the communication client requests an authentication using the authentication method that was agreed upon at the time of the B01 'request using of services API'.

5.2.4 PR4 (e-document transmitting process)

5.2.4.1 General

This process is for transmitting e-documents to the relying party using a trusted method and is regarded as a critical process on trusted communication of e-documents. After an entity gets a success as a result of 'communication authentication', the process is considered as a success. This process is mainly composed of two stage transactions called 'a) acquisition of receiving client information' and 'b) transmission of e-document and creation of TCE'.

On each stage transactions are as follows.

a) Stage: acquisition of receiving client information

Prior to the transmitting client sending e-documents to the receiving client, there is the stage of acquiring information about the receiving client, the intended recipient. Information about the receiving client was requested by the transmitting client or the transmitting server. If the transmitting client seeks to encrypt any e-documents or confirm the recipient on its own, then the transmitting client is to make such a request. However, if the transmitting client requests the transmission of an e-document including encryption of e-documents or confirmation of the recipient in its entirety over the transmitting server, then the transmitting server is to make such a request.

NOTE The process in 5.2.4.2 can be described as a process where a transmitting client acquires information about a receiving client before it is transmitted to a transmitting server. The process in 5.2.4.3 can be described as a process where the transmitting server that receives a transmission request from the transmitting client acquires information about the receiving client before immediately sending the e-document to the receiving server.

b) Stage: transmission of e-document and creation of TCE

This is the stage where TCE is generated to confirm the transmission and reception of an e-document by the communication server participated in trusted communication after the transmitting server has sent the e-document based on the information of the receiving client acquired.

5.2.4.2 PR4-1 (e-document transmitting process - basic type)

This process confirms the e-identity of the receiving client by retrieving information about the receiving client prior to the transmitting client sending the e-document. Also, before the transmitting client orders a transmission of the e-document for the transmitting server, it encrypts the e-document with the encryption key acquired from the receiving client which is used to reinforce its confidentiality. Figure 6 shows the overall feature of PR4-1.

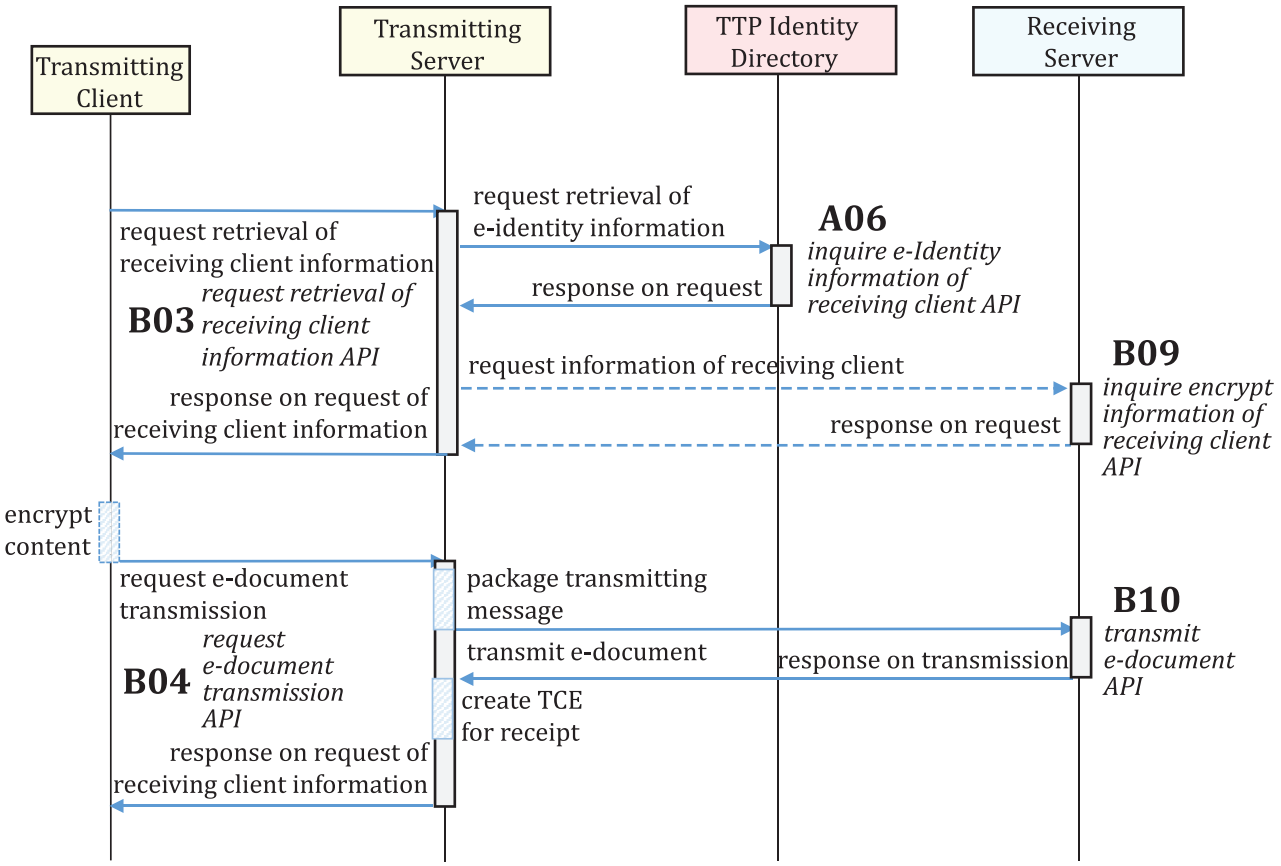


Figure 6 — PR4-1 (e-document transmitting process – basic type flow)

a) Related APIs

This process consists of B03 'request retrieval of receiving client information' API, B04 'request e-document transmission' API provided to the communication client by the communication server, A06 'inquire e-identity information of receiving client' API provided to communication server by TTP identity directory, B09 'inquire encrypt information of receiving client' and B10 'transmit e-document' API provided to the relying communication server.

b) Flow of process

This process consists of 2 stages, 1) the stage where the transmitting client acquires information about the receiving client, 2) the stage where the transmitting client sends the e-document via a transmitting server based on the information acquired by the transmitting client.

1) Stage: acquisition of receiving client information

i) Prior to a request being made for the transmission of the e-document, the transmitting client requests the transmitting server to obtain information of the receiving client in order to acquire the secure information needed for the confirmation about the receiving client and encryption of the transmitted document (call B03 'request retrieval of receiving client information' API).

ii) At the request of the transmitting client, the transmitting server will request and acquire information about the receiving server which acts as the agent for the receiving client over the identity directory (call A06 'inquire e-identity information of receiving client' API).

iii) Using the receiving server information acquired beforehand the transmitting server requests information about the receiving client (secure information for encryption, etc.) from the receiving server and then returns the acquired information to the transmitting client (call B09 'inquire encrypt information of receiving client' API). This is an optional stage. When additional information is required such as that the transmitting client requests an encryption key in order to encrypt the e-documents by itself before transmission of e-documents, this stage is implemented.

2) Stage: transmission of the e-document and creation of TCE

i) The transmitting client confirms the acquired information about the receiving client. If encryption for e-documents is necessary, then the e-documents are encrypted using the receiving client's (or receiving server's) encryption key. Whether or not the e-document will be encrypted is at the discretion of the transmitting client.

ii) Now the transmitting client requests to transmit the encrypted or not-encrypted e-document to the transmitting server (call B04 'request e-document transmission' API). In this stage, the transmitting client calls this API after adding the information required for NRO (non-repudiation of origin, refer to the ISO 19626-1:2020, 6.1) into a request message to be sent to the transmitting server.

iii) The transmitting server that receives a request for e-documents transmission from the transmitting client packages the e-document into a request message. And then the transmitting server transmits it to the receiving server according to the agreed protocol and receives the transmission results (call B10 'transmit e-document' API). In this stage, the transmitting server calls this API after adding the information required for NRS (non-repudiation of submission) into a request message to be sent to the receiving server. Once the receiving server has successfully received the request message, the information required for the NRR/NRD (non-repudiation of receipt/non-repudiation of delivery) is extracted from this message. After that, the receiving server adds a receipt notification ACK including this information to a response message and returns it to the transmitting server.

iv) And the transmitting server returns the response received from the receiving server to the transmitting client.

5.2.4.3 PR4-2 (e- document transmitting process - server delegation type)

This is the process where the transmitting client delegates the transmission of e-documents to the transmitting server fully. Prior to transmitting e-documents, the transmitting server that receives this transmission request retrieves information about the receiving client to confirm the accuracy of the e-identity of the receiving client. Also, if the transmitting server receives request of e-documents encryption from a transmitting client, prior to transmitting the e-documents, the transmitting server shall acquire an encryption key provided by the receiving client (or receiving server) and encrypt the e-documents.

Figure 7 shows the overall feature of PR4-2.

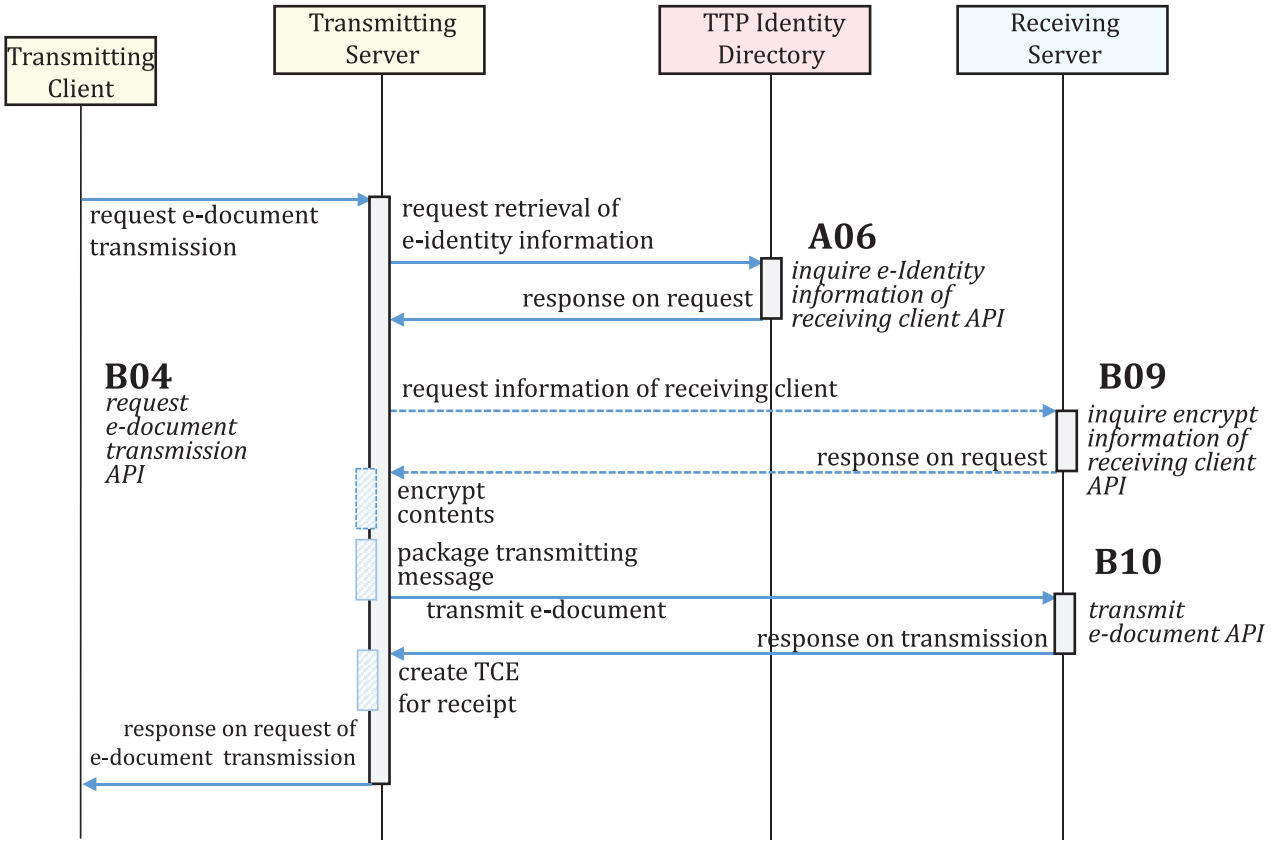


Figure 7 — PR4-2 (e-document transmitting process – server delegation type flow)

a) Related APIs

This process consists of B04 'request e-document transmission' API provided to the communication client by the communication server, A06 'inquire e-identity information of receiving client' API provided to the communication server by the TTP identity directory and B09 'inquire encrypt information of receiving client' and B10 'transmit e-document' API provided to the relying communication server.

b) Flow of process

This process basically consists of only one stage where the transmitting server sends the e-document according to a request of the transmitting client. Unlike the PR4-1 (process - basic type) that was proactively implemented for each of the necessary stages of e-document transmission by the transmitting client, with the PR4-2 (process - server delegation type) the transmitting server acquires information about the receiving client and directly processes the transmission of the e-document based on this information by itself.

Stage: transmission of the e-document and creation of TCE

1) The transmitting client requests the transmission of the e-document to the transmitting server. In this stage, the transmitting client calls the API after adding the information required for NRO (non-repudiation of origin, refer to the ISO 19626-1:2020, 6.1) and an option whether encrypting e-documents or not into a request message to be sent to the transmitting server (call B04 'request e-document transmission' API).

2) The transmitting server requests the TTP identity directory for the receiving client's basic information. Here the necessary basic information means the network information of the receiving server necessary for receiving the e-document instead of the receiving client from the capacity of an agent (call A06 'inquire e-identity information of receiving client' API).

3) Using the information of receiving server acquired beforehand the transmitting server requests additional information about the receiving client such as a security information needed for encryption (call B09 'inquire encrypt information of receiving client' API). This is an optional stage and as is the case when the transmitting client requests transmission after encrypting the e-document, this stage is implemented when additional information is required.

4) The transmitting server confirms the acquired information about the receiving client. If encryption for e-documents is necessary, the e-documents are encrypted using the receiving client's (or receiving server's) encryption key. Whether or not the e-documents will be encrypted is at the discretion of the transmitting client.

5) The transmitting server packages the e-documents into a request message. Then the transmitting server transmits it to the receiving server according to the protocol agreed on within the TCP and receives the transmission results (call B10 'transmit e-document' API). In this stage, the transmitting server calls this API after adding the information required for NRS (non-repudiation of submission) into a request message to be sent to the receiving server. Once the receiving server has successfully received the request message, the information required for the NRR/NRD is extracted from this message. After that, the receiving server adds a receipt notification ACK including this information to a response message and returns it to the transmitting server.

6) And the transmitting server returns the response received from the receiving server to the transmitting client.

5.2.5 PR5 (perusal confirmation process)

This is an optional process generated only if the transmitting client has requested 'perusal confirmation' while transmitting an e-document to the receiving client.

If examined in stages, this is according to the following procedures.

If a transmitting client has requested 'perusal confirmation' of the receiving client in the stage of transmitting an e-document, receiving server shall notify of this to the receiving server when receiving client views the message. The time of viewing the message by the receiving client is normally considered as the time when the client has requested the detailed information inquiry of the received message.

Once the behaviour of perusal by the receiving client is confirmed, the receiving server shall deliver to the transmitting server a confirmation signal of perusal instantly.

After the transmitting server has received the confirmation signal of perusal, it shall create TCE for perusal as legal and technical evidence on the fact that the receiving client has viewed the transmitted message. Figure 8 shows the overall feature of PR5.

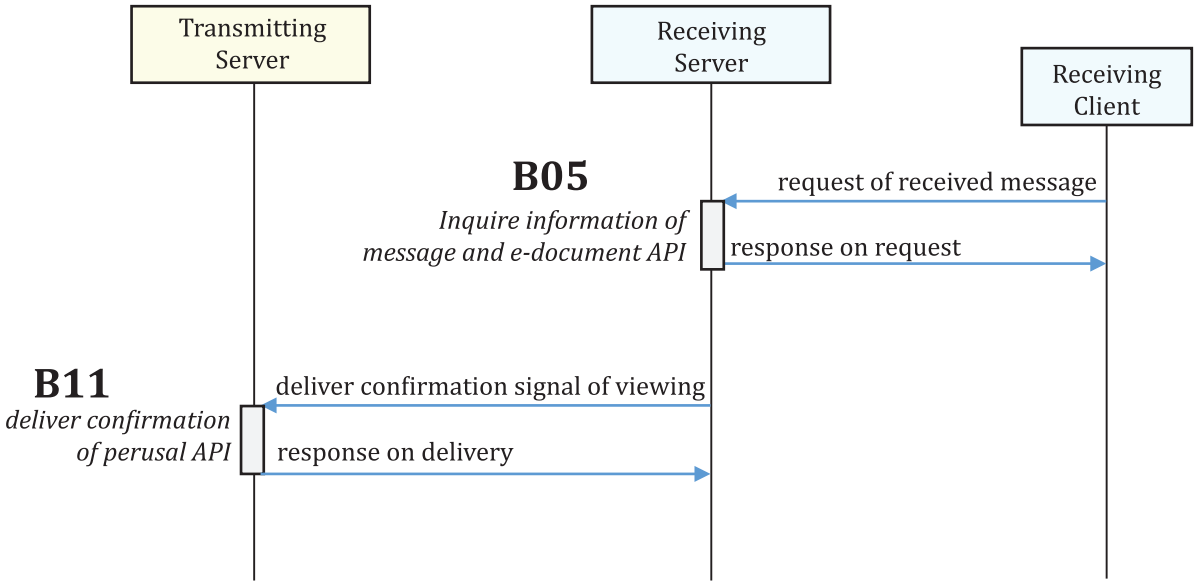


Figure 8 — PR5 (perusal confirmation process)

a) Related APIs

This process consists of B05 'inquire information of message and e-documents' API provided by the communication server to the communication client, and B11 'deliver confirmation of perusal' API provided to the relying communication server.

b) Flow of process

1) The receiving client requests detailed information of the received message and the attached e-documents to the receiving server in order to review the content of the received message (call B05 'inquire information of message and e-documents' API). Once the receiving server successfully replies to this request, this stage would be marked as a confirmation of perusal action.

2) Once the receiving server successfully replies with the detailed information of the message to the receiving client, the receiving server considers that the transaction is reading action performed by the receiving client and immediately transmits a signal (i.e. a perusal notification ACK) including perusal time, etc. to the transmitting server (call B11 'delivery confirmation of perusal' API).

3) If transmitting server receives a signal for confirmation of perusal from the receiving server, and then the transmitting server creates TCE for perusal confirmation including information in this signal after checking the validity of this signal.

5.2.6 PR6 (TCE preservation process)

After each communication server generates TCE for receipt or perusal and stores TCE in the TCE repository to keep it safe from hacking. This process has a pair of request-response to store TCE in the TCE repository. After being stored this way, TCE can later be searched or used when a confirmation or verification for the communication of e-documents is required.

Figure 9 shows the overall feature of PR5.

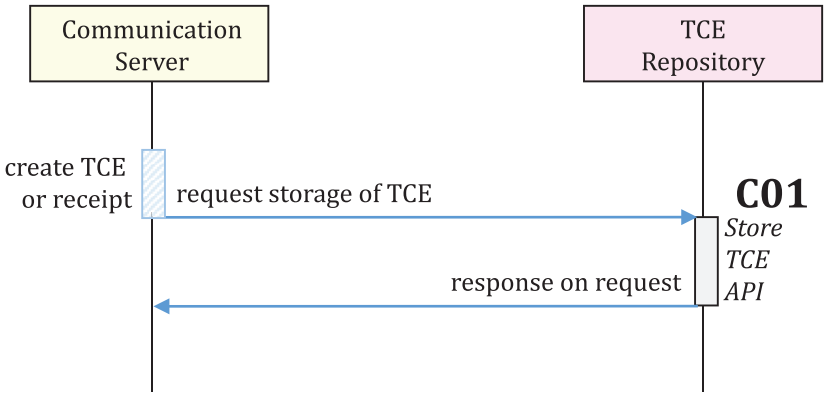


Figure 9 — PR6 (TCE preservation process)

a) Related APIs

This process consists of C01 'store TCE' API provided by the TCE repository to the communication server.

b) Flow of process

1) Once PR4 (e-document transmitting) or PR5 (perusal confirmation) has been successfully completed, the communication server will generate a TCE for receipt or perusal.

2) TCE generated by each communication server will undergo a validation before being stored in the TCE repository (call C01 'store TCE' API). The two communication servers' method for verifying TCE is to be decided by the TCP participants reaching an agreement on the matter.

5.2.7 PR7 (communication verification process)

This process has an API to request verification on the behaviour of transmitting or receiving e-documents.

When some disputes occur on the behaviour communicating e-document, someone requests verification on the behaviour and related information (transmission/reception time, originator/addressee and originality of delivered document, etc.) through this process.

The TCE repository that has received a request of verification on the behaviour of transmitting/receiving e-documents shall verify the behaviour in detail (transmission/reception time, originator/addressee and originality of delivered document, etc.) based on TCE stored on its own. Figure 10 shows the overall feature of PR7.

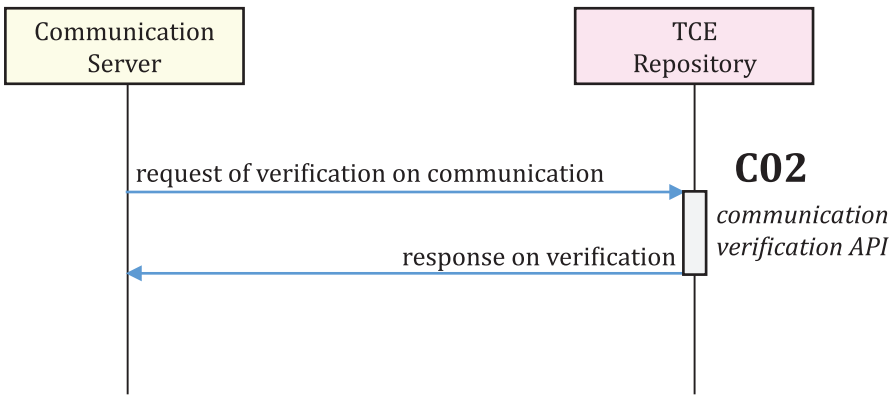


Figure 10 — PR7 (communication verification process)

a) Related APIs

This process consists of C02 'communication verification' API provided by the TCE Repository to the communication server and the communication client.

b) Flow of process

1) If the communication server or communication client requests a verification for the behaviour of communication with unique ID of TCE, then the TCE Repository shall verify the transmission/ reception time, originator/addressee and originality of the delivered document using TCE having the TCE ID and reply with the result (call C02 'communication verification' API).

5.2.8 PR8 (spam message handling process)

This process is the one to notify the result by determining the spam status after the receiving client reports this to the TTP identity directory and the TTP identity directory examines the reported message if the received message is determined as a spam as a process to prevent spreading a spam message under a TCP system.

If determined as a spam message by the TTP identity directory, the concerned transmitter shall not transmit e-documents under a TCP system by getting included on the blacklist. If this is examined in stages, the receiving client reports the spam message first and if determined as a spam message after the identity directory reviews the reported details, this is notified to each communication server participating in the TCP after including the e-identity of the transmitting client on the blacklist.

After the communication server manages the blacklist received from the TTP identity directory, the use is restricted so that the communication client using the concerned e-identity shall not transmit messages. Also, if the transmitting client of the received message falls under the blacklist, the receiving server rejects the reception of the concerned message. Figure 11 shows the overall feature of PR8.

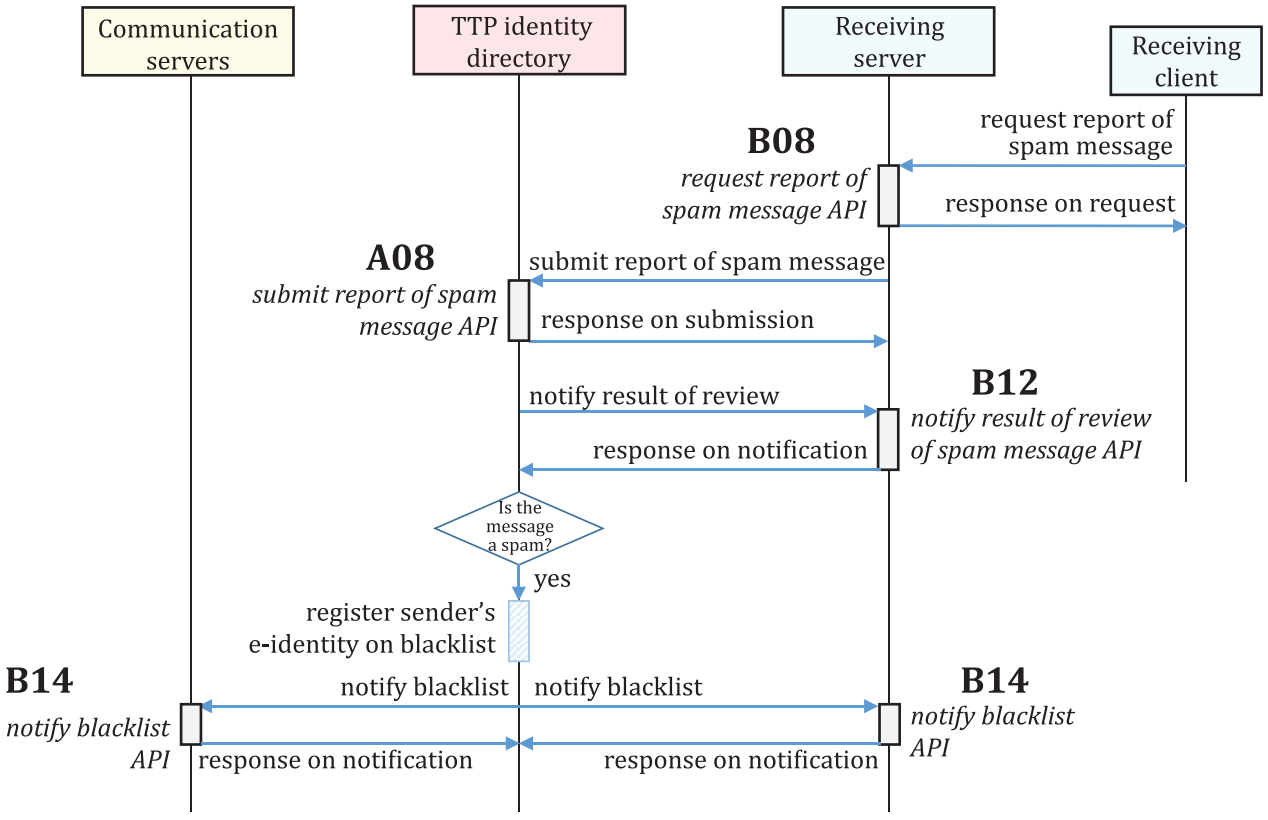


Figure 11 — PR8 (spam message handling process)

a) Related APIs

This process consists of B08 'request report of spam message' API provided by the communication server to the communication client, A08 'submit report of spam message' API provided to the communication server by the TTP identity directory, B12 'notify result of review of spam message' API and B14 'notify blacklist' API provided to the TPP identity directory by the communication server.

b) Flow of process

1) If the communication client deems the message received to be a spam message, this message will be reported to the communication server as a spam message (call B08 'request report of spam message' API).

2) The communication server then transmits the message reported as a spam by the communication client to the TTP identity directory (call A08 'submit report of spam message' API).

3) After reviewing the reported message, the TTP identity directory will decide whether the message is a spam or not.

4) The TTP identity directory will transmit the review results of the message to the communication server that the message was deemed to be a spam (call B12 'notify result of review of spam message' API).

5) If the TTP identity directory deems the reported message to be a spam message, the originator of this message will be added to a blacklist. After that, the TTP identity directory shall broadcast the revised blacklist to all participating in the TCP (call B14 'notify blacklist' API).

6 TCP APIs

6.1 General

The four components under a TCP system ('TTP Identity directory', 'TCP communication server', 'TCP communication client' and 'TCE repository') shall exchange some information by linking with one another for trusted communication in distributed environment.

6.2 Network requirements for APIs

6.2.1 General

Each system component shall provide the APIs necessary for linking with other systems and all APIs for exchanging information shall satisfy the following requirements.

— In case of linking each component, each API shall use the protocol to guarantee confidentiality of network, authentication on the relying party of communication and integrity on exchanged information for safety and reliability of exchanging information.

— Each component shall share the mutually agreed protocol in one TCP system and shall be linked based on this protocol.

6.2.2 Security requirements

6.2.2.1 General

In TCP, the security items required when interfacing between each component should satisfy the four types of conditions including guarantee of confidentiality on the exchanged information, authentication on the relying party of communication, guarantee of integrity on the exchanged information and non-repudiation on the behaviour of transmitting or receiving.

6.2.2.2 Confidentiality

Confidentiality of exchanging information needs to be performed from two types of perspectives. The first requirement is guarantee of confidentiality through network security applied in the communication protocol. The second requirement is the contents encryption targeting the information which needs confidentiality during exchange of messages. Although the first requirement 'confidentiality on network' is a mandatory item on all linked interfaces, the second requirement 'confidentiality of contents' via encryption' is an optional item applied by the policy of confidentiality is necessary on the contents at the time of transmitting the e-document by the transmitting client.

a) Confidentiality on network

For the confidentiality of network, SSL (Secure Socket Layer) or TLS (Transport Layer Security Protocol) is universally applied and used as a method encrypting all messages exchanged in the transport layer. Although which version of SSL or TSL to use may be determined by agreement between the TCPSPs, since 'confidentiality on network' is a mandatory requirement which is not optional inside the TCP, this shall be applied in case of the linkage through all interfaces.

SSL or TLS shall be applied on all of the following interfaces for the linkage between components.

— Main1 linkage interface (between a communication server and a TTP identity directory),

— Main2 linkage interface (between a communication server and a relying communication server),

— Main3 linkage interface (between a communication server and a TCE repository), and

— Client linkage interface (between a communication server and a communication client)

As SSL or TLS is the security which guarantees confidentiality between the two end-points of interfaces, decryption takes place at the initial stage of receiving the message on the data encrypted at the last stage of transmitting the message. Such a security method has an advantage of being able to be introduced easily since the end-to-end confidentiality can be guaranteed just by the configuration of SSL or TLS during the system configuration without separate consideration of security when developing the application.

On the contrary, there is a problem if confidentiality of the delivered message is necessary up to the component in the middle stage, since encryption and decryption are repeated by each stage if the transmission of message takes place when two or more interfaces become combined; such a security method is the one to guarantee confidentiality only between two contact points.

For instance, if the transmitting client delivers an e-document to the receiving client, the transmitting server and receiving server transmitting messages by the agent will also be able to check the details of e-document if only SSL or TLS is applied.

The part indicated as 'a) Confidentiality on network' in Figure 12 shows the section where encryption on the message is maintained when SSL or TLS is applied.

If the transmitting client desires to prevent the communication server at the middle stage from viewing the contents of e-document and only have the receiving client view the contents of e-document, security on the contents should be applied.

The part indicated as 'b) Confidentiality of contents' in Figure 12 shows the section where encryption on e-document is maintained when encryption is applied on the contents.

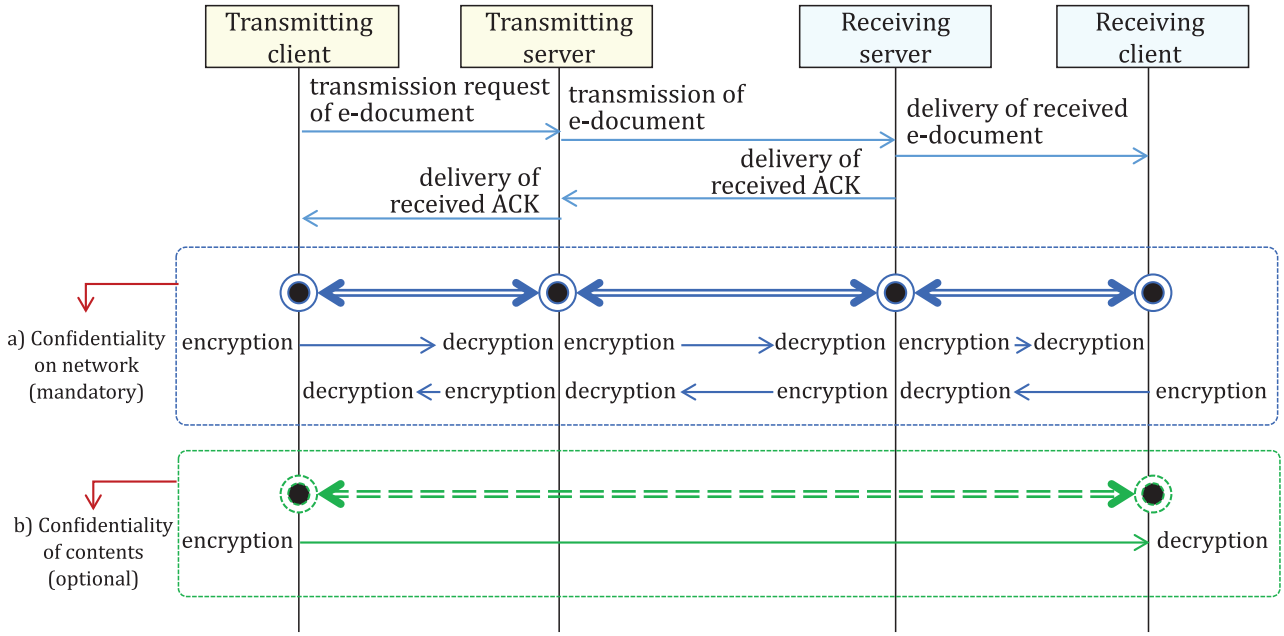


Figure 12 — Security area

b) Confidentiality of contents

In TCP, the process which needs confidentiality on the contents is PR4 (e-document transmitting process) and additional encryption of contents is not considered in other processes.

After performing the operation of encryption in the stage prior to the transmission by receiving the encryption key from the relying party, the encrypted document shall ensure confidentiality of contents in the process of transmitting the message.

For the method of encrypting the e-document in PR4 (e-document transmitting process), see 5.2.4.

6.2.2.3 Authentication on the relying party of communication

The message sender and message receiver in the interface between components shall perform mutual authentication on the relying party. To do it so, it should verify whether the other party is a legitimate entity through the process of handshaking in the SSL or TLS applied for the 'confidential on network'. Server certificate necessary in the handshake process shall be distributed by inputting into the whitelist.

While all mutual authentications on the relying party of communication through SSL or TLS shall be performed on the following interfaces, authentication on the server shall be basically performed in the interface between a communication server and a communication client.

— Main1 linkage interface (between a communication server and a TTP identity directory),

— Main2 linkage interface (between a communication server and a relying communication server),

— Main3 linkage interface (between a communication server and a TCE repository), and

— Client linkage interface (between a communication server and a communication client)

6.2.2.4 Guarantee of integrity on exchanged information

In order to guarantee integrity on exchanged information, it shall deliver by inputting hash information to guarantee integrity of content into the message header from the interface between components.

This document specifies the method of transmitting by inputting the digital signature on the message header with the certificate of sender after including the hash value guaranteeing the integrity of content.

After extracting the message header to verify integrity between hash information included in the header and the actually received contents, the message receiver shall verify whether the digital signature value of the header and information of the certificate used for the signature belong to the sender's one.

This method of verifying the integrity of exchanged information by inputting digital signature on the message header shall be applied mandatorily in Main2 linkage interface (between a communication server and a relying communication server).

To make it practicable, the communication server shall register the certificate for the digital signature of message as well as the certificate of server for SSL/TLS to the TTP identity directory. Also, the TTP identity directory shall distribute the whitelist by including the certificate information on each server.

6.2.2.5 Non-repudiation of communication

Under a TCP system, TCE performs a critical role of proving the fact of transmitting or receiving e-documents between communication clients and preventing repudiation of a communication between the originator and the addressee. For implementation of TCE, the communication servers shall create TCE by recording information of non-repudiation related to the fact of communication and store it into the TCE repository in a trusted method.

In TCE, there are two types: TCE for receipt confirmation to prove the fact of receiving e-document and TCE for perusal confirmation to prove the fact of perusing the e-document which the receiving client has received.

— TCE for receipt confirmation:

  — TCE for receipt confirmation is created in order to prove the fact of receiving the e-document properly by the receiving server after transmission.

  — The transmitting server creates TCE for receipt confirmation by combining the receipt notification ACK received from the receiving server and the basic information to classify the TCE.

— TCE for perusal confirmation:

  — The transmitting client requests a perusal notification ACK to the receiving server through the transmitting server in case of checking the state of the receiving client's perusal. Therefore, TCE for perusal confirmation is optional evidence material created only if there is a request of perusal notification from the transmitting client.

  — The receiving server that has received this request shall deliver a perusal notification ACK to the transmitting server when the receiving client peruses the received e-document.

  — The transmitting server creates TCE for perusal confirmation by combining the perusal notification ACK received from the receiving server and the basic information to classify the TCE.

See Annex A for a detailed structure of TCE.

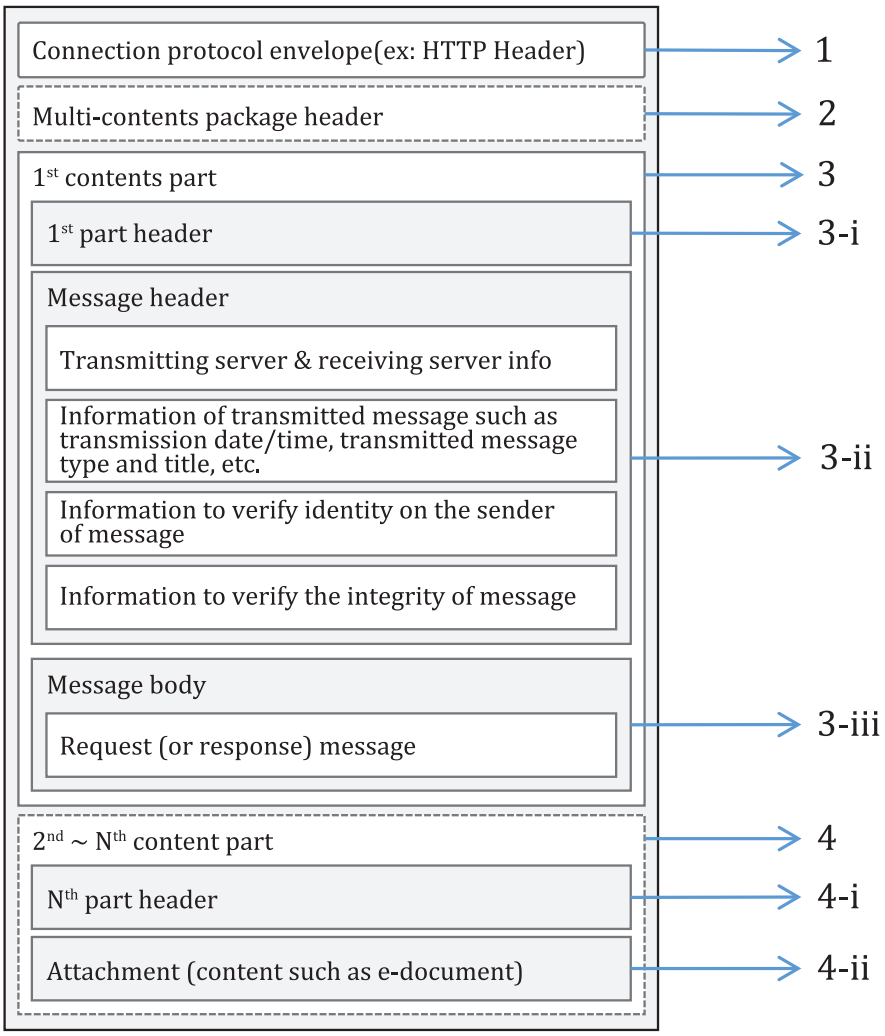
6.2.3 Common requirements for protocol

a) Structure of message package

The message exchanged for linking between each component is composed of multiple message containers.

The first message container is composed of the message header containing basic information on the transmitted or received message and the message body area containing context on the message delivered to the sender or the recipient as an essential item of all messages.

The message containers after the second one shall be extended as much as needed by inputting one attached file into one container as an optional item which exists only if there is an attached file. Figure 13 shows a structure of message package.



Key

1 connection protocol header
2 header for multi container envelope
3 1st Message container envelope
3-i header for 1st message container envelope
3-ii message header
3-iii message body
4 Nth Message container envelope
4-i header for Nth message container envelope
4-ii attachment content

Figure 13 — Structure of message package

b) Details of message structure

Each item which composes a message is as follows.

1) Connection protocol header (mandatory)

The header information basically defined in the protocol is described here when the entire message becomes enveloped according to the communication protocol agreed for the linkage between system components inside the TCP.

2) Header for multi-container envelope (optional)

If there is additional data (document) attached in addition to the basic message when linking system components, a protocol for packaging multi-contents is necessary in order to deliver a number of contents. In this part, the header information necessary according to the packaging for multi-contents shall be described; and this part can be omitted if there are no attached files.

3) 1st Message container envelope (mandatory)

As a container which contains the first content within the message, this part is composed of the header information to classify the container, the message header commonly included in all messages to be delivered, and message body describing what the details of the concerned message are.

i) Header for 1st message container envelope

In this part, the separator for the first container out of the entire message and the basic information on the message (message type or message length, etc.)

ii) Message header

Message header is configured by inputting in common items except for business details on the message exchanged between a transmitting server and a receiving server, for instance the information such as e-identity ID of transmitting client, e-identity ID of receiving client, message type or message title. See Annex B for a detailed structure of the message header.

iii) Message body

As this part contains the business-related details which the sender of the message intends to send to the recipient, request item is contained in the request message and response item or error item is contained in the response message. Table 1 describes a structure of message body.

Table 1 — Structure of message body

Name of element	Description	Notes
MessageBody	— Root element of request message	
RequestMSG (ResponseMSG, ErrorMSG)	— Appropriate request, response or error message is described depending on each interface.	

4) N^{th} Message container envelope (optional)

This part delivers attached files (documents) in order from the second container as many as the number of containers putting an attached file if there are any attached files in a delivering message. In this container, information of the header separating the container and the attached files (documents) to be delivered are put in.

i) Header for N^{th} message container envelope

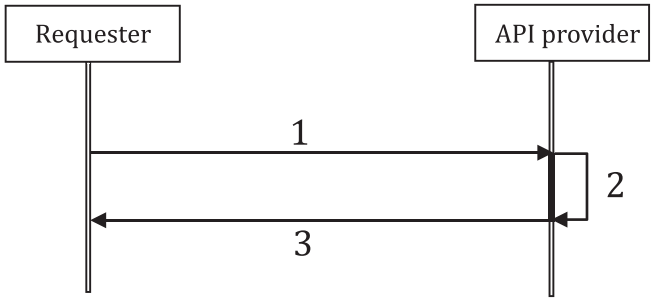
In this part, the separator for separating each container and the basic information on the contents (message type or message length, etc.) are described.

ii) Attachment content

In this part, the actual content to be delivered, in other words the attached file (document) is put in to be delivered.

c) Message exchange pattern

The APIs providing main components within the TCP exchange messages following a request-response pattern as shown below. Figure 14 shows the message exchange pattern of API.



Key

1 inquire some information or request some processing to API provider

2 query some information or execute some process

3 return a result of inquiry or execution

Figure 14 — Message exchange pattern of API

The API is provided by each component in the TCP and is performed in the following steps. At first a requester sends a request message to the API provider, then the API provider performs an internal process according to this request, and as a final step the API provider returns a response message synchronously. The main API within the TCP shall clearly identify the success or failure status of each transaction in order to ensure reliability of the message exchange. Therefore, a pattern following a synchronous request-response method is requested.

Of course, aside from the API mentioned in this document, additional API provided for TCP to offer more diverse services has a unique message exchange pattern for each API.

6.3 Requirements for service interface

6.3.1 APIs of TTP identity directory

Table 2 describes an API list of the TTP identity directory.

Table 2 — API list of TTP identity directory

No.	Name of API	Description	Linkage target
A01	Register communication server	— API to register information on the communication server participating in the TCP	Communication server
A02	Inquire information of communication server	— API to get information of registered communication server.	Communication server or communication client
A03	Manage information of communication server	— API to modify or remove the concerned information if the information of the communication server registered to the identity directory gets changed or if the removal of server information is necessary	Communication server
A04	Identify entity	— API to verify whether the information presented by the entity and the actual owner of the concerned information are matching in the process of registering an e-identity to participate in the distribution of trusted electronic document by the entity	Communication server or communication client
A05	Register e-identity	— API to register an e-identity to uniquely identify the entity in the TCP in order to participate in the distribution of trusted electronic document by the entity	Communication server or communication client
A06	Inquire e-identity information of receiving client	— API to get registered e-identity information — According to a type of a requester, the requester gets a different scope of data from TTP identity directory.	Communication server or communication client
A07	Manage information of e-identity	— API to modify information when the e-identity information registered by the entity has been changed or to remove information when the entity no longer uses the registered e-identity	Communication server or communication client
A08	Submit report of spam message	— API to report the received message as a spam message when it has been determined as a spam message	Communication server
A09	Retrieve whitelist	— API to request an inquiry of the whitelist Information registered in the identity directory	Communication server
A10	Retrieve blacklist	— API to request an inquiry of the blacklist Information registered in the identity directory	Communication server

See C.1 for more information on each API.

6.3.2 APIs of communication server

Table 3 describes an API list of the TCP communication server.

Table 3 — API list of TCP communication server

No	Name of API	Description	Linkage target
B01	Request using of services	— API for communication client to request use of the services provided by communication server	Communication client
B02	Authenticate communication client	— API for the communication client to request authentication in order to use the functions(services) provided by the server by connecting to the communication server	Communication client
B03	Request retrieval of receiving client information	— API for the communication client to request retrieval of receiving client information to the communication server	Communication client
B04	Request e-document transmission	— API for the communication client to request transmission to the communication server in order to transmit an e-document to the recipient	Communication client
B05	Inquire list of transmission / reception messages	— API for the communication client to request inquiry of list to the communication server in order to inquire the list of transmitted / received messages	Communication client
B06	Inquire information of message and e-documents	— API for the communication client to request detailed information to the communication server in order to inquire detailed information on transmitted / received messages	Communication client
B07	Request message deletion	— API for the communication client to request to delete the messages stored in the communication server	Communication client
B08	Request report of spam message	— API for the communication client to request report a specific message as a spam message to the communication server	Communication client
B09	Inquire encrypt information of receiving client	— API for the transmitting communication server to request inquiry on whether the account information of the recipient is valid to the Receiving communication server	Communication server
B10	Transmit e-document	— API for the transmitting communication server to transmit e-document to the receiving communication server	Communication server
B11	Deliver confirmation of perusal	— API for the Receiving communication server to deliver a confirmation letter to notify receipt or perusal after receiving an e-document	Communication server
B12	Notify result of review of spam message	— API to notify the reporter of the review results after reviewing the spam message status based on the details of reporting as a spam message	TTP identity directory
B13	Notify whitelist	— API to notify the communication server participating in the TCP when the information of whitelist has been changed	TTP identity directory
B14	Notify blacklist	— API to notify the communication server participating in the TCP when the information of blacklist has been changed according to the spam message review result, etc.	TTP identity directory

See C.2 for more information on each API.

6.3.3 APIs of TCE repository

Table 4 describes an API list of the TCE repository.

Table 4 — API list of TCE repository

No	Name of API	Description	Linkage target
C01	Store TCE	— API for a communication server to request the storage of TCE generated as a communication confirmation received to the TCE repository	Communication server
C02	Communication verification	— API for communication server or communication client to request to verify the behaviour of communication that occurred in the past to TCE repository	Communication server or communication client

See C.3 for more information on each API.

Annex A
(informative)

Structure of TCE

A.1 General

If an originator transmitted e-documents on the trusted communication platform to an addressee, a TCE is generated to verify that the e-documents were sent by the originator, or that the addressee received it, or that the addressee read the e-documents. To verify whether the e-documents were sent or received, a TCE for receipt confirmation is created, and to verify that the received e-documents were perused, a TCE for perusal confirmation is created.

The TCE is an evidence of an action of receipt or perusal; therefore, to verify this, information about the act of sending, receiving and perusal and information about the originator and addressee, and information about the transmitted e-documents shall be included. Of course, depending on the business model regarding the purpose of the TCP or the agreement made among the participants, additional information can be required, but the basic structure that a TCE should bring is described in A.2.

A.2 Structure of TCE

A TCE is created by the following structure including receipt notification or receipt notification ACK and the basic information on the TCE sent by the recipient in order to record the fact of delivering e-document in a reliable way.

The TCE is uniquely identified by combination of information such as TCE identifier, TCE type and communication ID out of the information inside TCE. Figure A.1 shows the structure of a TCE.

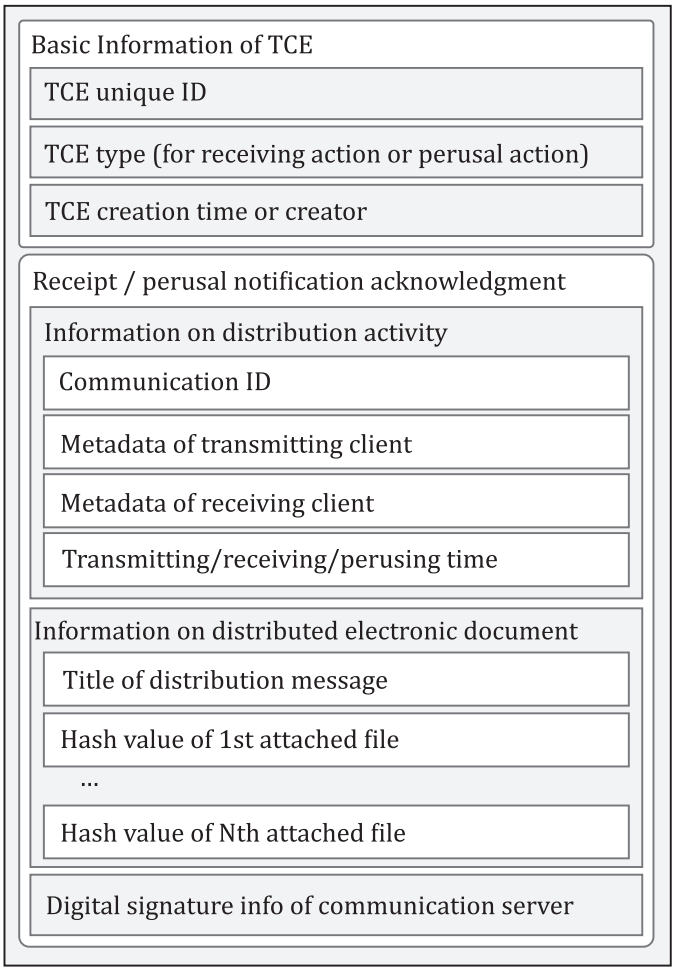


Figure A.1 — Structure of TCE

A.3 Basic information of TCE

This part is composed of the basic information to identify the TCE. Table A.1 describes the structure of basic information in a TCE.

Table A.1 — Structure of basic information in TCE

Element name		Description	Note
TCEInfo		Basic information on the TCE	
	TCEIdentifier	The identifier for identifying the TCE in which the TCE is created	Name or alias which represents the TCE when TCP is created in country, community or business group units
	TCEType	The value to identify which activity the evidence is for in trusted communication	'01': Created for the purpose of proving that the receiving server has received the e-document properly as a TCE for receipt notification '02': Created for the purpose of proving that the receiving client has perused the e-document properly as a TCE for perusal notification
	TCECreation Time	The time on which the transmitting server has created the TCE	UTC format 2017-10-27T13:24:54.745Z

Table A.1 *(continued)*

Element name	Description	Note
TCEInfo	Basic information on the TCE	
TCECreator	Information of the creator who has created the TCE	Unique ID of the transmitting server

A.4 Signal for receipt notification or perusal notification

This part is the receipt notification ACK or perusal notification ACK which the transmitting server has received from the receiving server as it is. ACK signal for receipt or perusal notification is composed of information of communication, information of e-document received and digital signature signed by the communication server. Table A.2 describes the structure of an ACK signal.

Table A.2 — Structure of ACK signal

Element name	Description	Note
Acknowledgment	Receipt notification ACK or perusal notification ACK signal received from the receiving server	
CommunicationInfo	Information on the communication to distribute e-documents between transmitting server and receiving server	
CommunicationID	A unique ID for classifying the distribution unit of e-documents	As an ID value transmitted after being uniquely created inside the TCP at the time of transmitting an e-document by the transmitting server, transmitting and responding message become connected as one communication ID
TransmitterInfo	Information on Transmitter	
TransmittingClient	An e-identity ID value of the transmitting client	A unique e-identity ID for identifying the communication client performing the role of an originator in the TTP identity directory
TransmittingServer	A unique ID value of the transmitting server	An ID value to uniquely identify the communication server which actually transmits e-documents by the request of the transmitting client inside the TCP
ReceiverInfo	Information on receiver	
ReceivingClient	An e-identity ID value of the receiving client	A unique e-identity ID for identifying the communication client performing the role of an addressee in the TTP identity directory
ReceivingServer	A unique ID value of the receiving server	An ID value to uniquely identify the communication server which actually receives e-documents by the request of the receiving client inside the TCP
TransmittingTime	The time on which the transmitting server transmits an electronic document to the receiving server	UTC format 2017-10-27T13:24:52.855Z

Table A.2 *(continued)*

Element name	Description	Note
ReceivingTime	The time on which the receiving server has received an electronic document from the transmitting server	UTC format 2017-10-27T13:24:53.031Z
PerusingTime	The time on which the receiving client has viewed the electronic document received from the receiving server (An item included only in the TCE for viewing notification)	UTC format 2017-10-27T13:24:53.031Z
eDocumentInfo	Basic information on the electronic document distributed between transmitting server and receiving server	
TitleMSG	Title of the message to transmit an electronic document	
AttachedFileNo	Total number of the delivered e-documents	Total number of e-documents transmitted from one communication
AttachedFileInfo	Information on the delivered electronic document	Described repeatedly as many as the number described on the AttachedFileNo
SeqNo	Sequence of being attached to the message	
FileName	File Name of the electronic document to deliver	
FileHashInfo	Hash information on the delivered electronic document	
HashAlgorithm	Algorithm for extracting the hash value of File	
HashValue	Hash value of the File extracted according to the Hash algorithm	
DigitalSign	The digital signature information which the communication server has added in order to assign reliability on the information described in the acknowledgement signal on the fact of receiving or viewing an electronic document	

Annex B
(informative)

Structure of message header

Aside from the mere trusted communication of e-documents, in order that all participants in the TCP participate in trusted communication and that information related to trusted communication is managed, connections and communication between each component shall be made. To make such connections, the basic structure of messages required in the process of communication were provided (see 6.2.3). Of these, regardless of the purpose of the connection, within the message of communication, the recipient, basic information to identify the message, and digital signature information for securing reliability shall be included, but in the first contents part of the basic structure of the message such information should be included in the message header.

Table B.1 provide suggestions for the basic configuration of information that should be included in the header.

Table B.1 — Structure of message header

Name of element	Description	Notes
MessageHeader	— Root element of request message	-
From	— Information on the sender of message	-
PartyInfo	— Information on the system owner requesting the transmission of message	-
ID	— Unique ID value of the owner	String
Type	— Unique ID type of the owner — Business registration no. or corporation no., etc.	String
Role	— Role of the sender of message	String
To	— Information on the recipient of message	-
PartyInfo	— Information on the system owner receiving the message	-
ID	— Unique ID value of the owner	String
Type	— Unique ID type of the owner — Business registration no. or corporation no., etc.	String
Role	— Role of the recipient of message	String
CommunicationID	— Unique ID of business unit to accomplish one purpose — Performs the role of binding these messages as one when more than one message is transmitted / received for business purposes	String
CommunicationType	— Specifies the type of communication — Set up as ‘TCE_Comm’ as default in trusted communication — Utilized when extending as various communication types in the future	string
Service	— Unique name of business unit to accomplish one purpose	String
Action	— Refers to the purpose of message in the work	String
MessageData	— Unique information on message	-

Table B.1 *(continued)*

Name of element	Description	Notes
Version	Version of the protocol which the message complies with	String
MessageID	— Unique ID value of the message	String
RefToMessageID	— Describes the MessageID value on the requested message if the message is a response message	String
Description	— Description on the details of message	String
ExtendedNote	— Extended data region used if an entity has additional items that need to be delivered	-
Subject	— Name of the extended data	String
Content	— Data value	String
Contents_Info	— Information on the contents that need integrity (Described repeatedly as much as needed)	
Contents_ID	— Information to identify contents in the message	
Contents_hash	— Hash value on the contents	
DigitalSign_Info	— Digital signature information on the message	
Cert_Info	— Information of the certificate used for digital signature	
Algorithm_Info	— Information of the Algorithm used for digital signature	
DSig_Value	— Digital signature value on the message header except for the information on the DigitalSign_Info	

Annex C
(informative)

Detailed description for APIs

C.1 APIs of TTP identity directory

C.1.1 A01'register communication server' API

Server information shall be registered to the identity directory for the communication server to participate in trusted distribution under a TCP system. TTP identity directory that has received a registration request shall verify whether the concerned server is implemented by conforming to the standard protocol and whether it is interoperable with other component inside the TCP prior to the registration of server information, and only the verified servers shall be registered and included in the whitelist.

a) Related process

— PR1 (communication server registration process) (see 5.2.1).

b) Description of API

— Requester

  — The communication server is a requester that requests A01 'register the communication server' to the TTP identity directory.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes basic information of communication server: name, network address, public key for encryption and digital signature verification, etc.

    — And management information of communication server: contact information of owner or manager.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — After receiving the request message, TTP identity directory verity whether this server is the one that has completed verification on conformity of standards and interoperability in advance, and whether all necessary information has been submitted.

  — TTP identity directory registers it on the whitelist as a server that can participate in trusted communication.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process.

    - If the result is success then response message includes unique ID issued by TTP identity directory.
    - Else if the result is failure then response message includes error code, error description, etc.
- Attachment in 2nd ~ Nth content part
    - Nothing.

C.1.2 A02 'inquire information of communication server' API

This API is used to get information of communication server registered at TTP identity directory. Usually information of communication server is open to every participant.

a) Related process

- No related process.

b) Description of API

- Requester
    - The communication server is a requester that requests 'inquire information of communication server' to the TTP identity directory.
    - But TTP identity directory does not restrict retrieving information of communication server with any special authority.
- Structure of message sent by requester
    - Request message in message body of 1st content part
        - Includes the unique ID of communication server which received from TTP identity directory after registering it.
    - Attachment in 2nd ~ Nth content part
        - Nothing.
- Internal process
    - Retrieves information of communication server corresponding with the unique ID or name of communication server.
- Structure of message replied by responder
    - Response message in message body of 1st content part
        - Includes success or failure as a result of internal process.
        - If the result is success, then response message includes information (basic and management information) of the communication server.
        - Else if the result is failure then response message includes error code, error description, etc.
    - Attachment in 2nd ~ Nth content part
        - Nothing.

C.1.3 A03 'manage information of communication server' API

If the information of the communication server registered to the identity directory is modified or if the server is no longer participating (or unable to participate) as a communication server, the server shall request change of the concerned communication server information (modification or removal). The

TTP identity directory shall perform an operation to change the concerned information after verifying whether the requester who has requested the change of information is the requester who has authority to make changes on the communication server.

a) Related process

— Related process is not described in this document because this API is not used in main processes. But if the information of the communication server is changed or the communication server does not participate in TCP anymore, then this API is used with similar step to A01'register communication server' API in PR1 (communication server registration process) (see 5.2.1).

b) Description of API

— Requester

  — The communication server is a requester that requests A03 'manage information of a communication server' API to the TTP identity directory.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes request type: this request is whether 'modify' or 'delete'.

    — If the request type is 'modified,' the request message includes the information regarding the necessity of revision for the unique ID of the communication server that is to be revised.

    — If the request type is deleted, the request message includes the information regarding the unique ID of the communication server that is to be deleted.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — After receiving the request message, TTP identity directory shall perform an operation to change (or to delete) the concerned information after verifying whether the requester who has requested the change (or deletion) of information is the requester who has authority to make changes on the communication server.

  — If changed information of communication server is concerned with information for connection to the server or a registered communication server is deleted, then TTP identity directory applies the change to whitelist.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process.

    — If the result is failure then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.1.4 A04 'identify entity' API

TTP identity directory shall verify whether the online entity connected to the TCP through a communication client is the same one as the offline entity prior to the registration of the entity to use the trusted e-document communication.

In order to verify whether it is the same one or not, TTP identity directory verifies that the online entity is the legitimate owner of a characteristic information (e.g. resident registration number, social security number) submitted by the online one. Communication client or communication server performs the verification of entity identity using this API provided by the TTP identity directory in order to register the entity to the TTP identity directory.

a) Related process

— 'PR-2 (e-identity registration process)', 'modification or deletion information of e-identity process' (this process is not described in this document because it is not used in a main process, this process consists of this API and A07 'manage information of e-identity' API).

b) Description of API

— Requester

  — The communication server or communication client is a requester that requests authenticate identity to the TTP identity directory.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes characteristic information that TTP identity directory can identify who the requesting entity is on offline space.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — TTP identity directory verifies whether the connected entity is a legitimate owner of the submitted characteristic information or not.

  — TCP's internal policy will decide what characteristic information the TTP identity directory will have and by what method of the entity's identification will be confirmed.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process.

    — If the result is success, then response message includes some information as an evidence of authentication success.

    — Else if failure, then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.1.5 A05 'register e-identity' API

For an entity to participate in the TCP, after authenticating, the entity shall register an e-identity to the TTP identity directory. If result of registering an e-identity is success, then ID (that is an e-identity ID) of the registered e-identity becomes a unique alias name to represent the entity in the TCP from now on and gets to refer to the entity who owns the e-identity.

a) Related process

— PR2 (e-identity registration process).

b) Description of API

— Requester

  — The communication server or the communication client is a requester that requests 'A05 register e-identity' API to the TTP identity directory.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes the token received as an evidence of authentication success from TTP identity directory.

    — Includes basic information of the entity: characteristic information, name.

    — Includes an e-identity ID which an entity wants to use as an alias name in TCP.

      — An e-identity ID should be unique in a TCP and naming rule of ID is decided by policy of TCP.

    — Includes management information of entity: information of contact point such as e-mail, address, phone no, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — TTP identity directory verifies whether the received message is valid or not.

  — If it is valid, then TTP identity directory registers an e-identity ID with the information submitted by requester.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process.

    — If failure, then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.1.6 A06 'inquire e-identity information of receiving client' API

This API is used to get e-identity information of receiving client registered at TTP identity directory. This API has two types of requests. One request type is network information of an e-identity as an addressee (a receiving client), the other one is basic and management information of the e-identity.

Searches for network information typically are performed before the transmitting client (or server) sends the e-document and is used to search network information about the addressee. Therefore, this search can be performed by any TCP participant as a requester. However, searches on the basic and management information of an e-identity are only permitted by users having an access authentication (owner of an e-identity and sometimes managers).

a) Related process

— If the request type is for network information, then the PR4 (e-document transmitting process) (see 5.2.4).

b) Description of API

— Requester

  — In case of searches for network information, generally the communication server is a requester that requests an A06 'inquire information of e-identity' API to the TTP identity directory.

  — For full information searches about the basic and management information about the e-identity, an authenticated entity would request an A06 'inquire e-identity information of receiving client' API to the TTP identity directory via the communication client or the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes a type of request: 'network information' or 'full information'.

    — Includes the ID of the e-identity that is the target of the search.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — For 'full information' type of request, prior to performing the search, the authorization would be performed to determine if the requester has an access authentication for the information of the e-identity. The information of the e-identity can include privacy information about the entity, therefore, an access control is required.

  — For 'network information' type of request, the information can be disclosed to all users participating in the TCP. Therefore, any authorization for this request is not performed.

  — If the requester has a legitimate authority for the information, then TTP identity directory retrieves access information of the e-identity corresponding with the e-identity ID submitted by the requester.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process.

    — If the result is success:

      — and type of request is 'full information', then response message includes full information (basic and management information) of the e-identity;

      — and type of request is 'network information', then response message includes a network information to send an e-document to the addressee having the e-identity;

      — 'network information' is the information for connecting to the communication server, as a proxy of the e-identity and so on, which will perform the transmission and reception of the e-document.

    — Else if the result is failure, then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.1.7 A07 'manage information of e-identity' API

This API is used in case the information of an e-identity registered to the TTP identity directory is changed or the e-identity does not participate in TCP anymore. Before modifying or removing information of the e- identity, the TTP identity directory handles the requested action after verifying whether the requester is an entity with a legitimate authority.

a) Related process

— Related process is not described in this document because this API is not used in main processes. But if the information of an e-identity is changed or the e-identity does not participate in TCP anymore, then this API is used with similar step to A05 'register e-identity' API in PR2 (e-identity registration process) except the step calling B01 'request using of service' API.

b) Description of API

— Requester

  — The communication server or communication client is a requester that requests 'manage e-identity' to the TTP identity directory.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes the token received as an evidence of authentication success from TTP identity directory.

    — Includes request type: this request is whether 'modifying' or 'deleting'.

    — If the request type is 'modified,' the request message includes the information regarding the necessity of revision for the e-identity ID that is to be revised.

    — If the request type is deleted, the request message includes the e-identity ID that is to be deleted.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — Verify whether the requester who has requested the change (or deletion) of information is the requester who has authority to make changes on the e-identity and requested message is valid or not.

  — If result of verification is valid, TTP identity directory shall perform an operation to change (or to delete) the concerned information.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process.

    — If the result is failure, then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.1.8 A08 'submit report of spam message' API

After the communication server receives a report about a spam message from the receiving client, this message will be reported to the TTP identity directory as a spam message. This API is used when the communication server reports the spam to the TTP identity directory.

a) Related process

— PR8 (spam message handling process) (see 5.2.8).

b) Description of API

— Requester

  — The communication server is a requester that requests A08 'submit report of message' API to the TTP identity directory.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes full content of a message considered as a spam message.

    — Includes the e-identity of the receiving client that reported this spam message.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — The message reported as a spam message to the TTP identity directory is registered to the list of spam messages pending verification to be verified by the system or manager.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of receipt of the submitted message.

    — If the result is success then response message is just ACK message.

    — Else if the result is failure, then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.1.9 A09 'retrieve whitelist' API

If a communication server does not have a whitelist because it is participating in the TCP for the first time, or because it has not received whitelists that are delivered by TTP identity directory due to a system abnormality or operation issue, then it is possible that the whitelist held is not its latest version. In this case, the communication server will not wait until the TTP identity directory sends one, but using this API, it can be received by requesting the latest whitelist to the TTP identity directory.

a) Related process

— No related process.

b) Description of API

— Requester

  — The communication server is a requester that requests A09 ‘retrieve whitelist' to the TTP identity directory.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes a request type (whether request of full latest whitelist or list of communication server changed since a requested date in whitelist).

    — If the request type is 'list of communication server changed since a requested date’, then includes the requested date in whitelist.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — Depending on the request type, the TTP identity directory either searches for the full latest whitelist or from among a list of communication server changes since a requested date.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of receipt of the submitted message.

    — If the result is success then response message includes a latest full whitelist or list of communication server changed since the requested date in the whitelist.

    — Else if the result is failure then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.1.10 A10 ‘retrieve blacklist’ API

If a communication server does not have a blacklist because it is participating in the TCP for the first time, or because it has not received blacklists that are delivered by TTP identity directory due to a system abnormality or operation issue, then it is possible that the blacklist held is not the latest version white list. In this case, the communication server will not wait until the TTP identity directory sends one, but using this API, it can be received by requesting the latest blacklist to the TTP identity directory.

a) Related process

— No related process.

b) Description of API

— Requester

  — The communication server is a requester that requests 'inquire blacklist' to the TTP identity directory.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes a request type (whether request of full latest blacklist or list of an e-identity changed since a requested date in blacklist).

    — If the request type is 'list of the e-identity changed since a requested date in blacklist', then includes the requested date.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — Depending on the request type, the TTP identity directory either searches for full latest blacklist or list of the e-identity changed since a requested date in the full blacklist.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of receipt of the submitted message.

    — If the result is success then response message includes a latest full blacklist or list of the e-identity changed since the requested date in the blacklist.

    — Else if the result is failure then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.2 APIs of Communication server

C.2.1 B01 'request using of services' API

The communication client as an entity's proxy shall request the use of the service to the communication server that acts as an agent for the actual trusted communication with e-documents for participation in the TCP, and receive an approval of the request. This is an API where the communication client requests the use of service to the communication server.

The service shall undergo an agreement stage regarding the use of service and the terms and conditions of use, therefore, aside from the information specified in this document, additional information and procedures are necessary. This will be decided according to the operating policy of the communication server. Here only the most basic, required categories are specified.

a) Related process

— PR2 (e-identity registration process)

b) Description of API

— Requester

  — The communication client is a requester that requests B01 'request using of services' API to the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes basic information of communication client: name, e-identity ID, etc.

    — And management information of communication server: contact information of the e-identity (such as the address, email address, phone number).

    — Additional information for the use of service requested by the communication server (for example, period of service use, service levels).

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — Communication server verifies that the communication client is a legitimate user having the e-identity ID.

  — Communication server decides based on the operation policy if the request of the service used by the communication client is approved.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes acceptance or rejection as a result of internal process.

    — If the result is rejection, then response message includes reason of rejection, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.2.2 B02 'authenticate communication client' API

A communication client shall obtain user authentication before using the services provided by the communication server. Communication server decides through this API if the communication client is registered as a service user at the communication server.

To do so, the communication server will use a wide range of authentication methods such as password authentication method, onetime password authentication method, and the biometrics method. This will be determined according to an agreement between the TCP participants and the policies of the communication server.

a) Related process

— PR3 (communication authentication process) (see 5.2.3).

b) Description of API

— Requester

  — The communication client is a requester that requests B02 'authenticate communication client' API to the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes an e-identity ID of the entity and information for authentication (such as password, information of finger print, information of entity's face).

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — The communication server implements user verification by using the e-identity ID and information for authentication included in the request message.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process.

    — If the result is success then response message includes authentication token.

    — Else if the result is failure then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.2.3 B03 'request retrieval of receiving client information' API

This API is used to acquire the network information of the receiving client prior to the communication client transmitting the e-document. In the process of transmitting the e-document, when the communication client adopts the method of confirming the receiving client's information or implementing encryption of e-documents by itself (5.2.4.2) rather than the method where the communication client requests confirming the receiving client's information or encryption of the e-documents to the communication server in its entirety (5.2.4.3), a 'request retrieval of receiving client information' API call is made to the communication server.

a) Related process

— PR4-1 (e-document transmitting process– basic type) (5.2.4.2).

b) Description of API

— Requester

  — The communication client is a requester that requests B03 'request retrieval of receiving client information' API to the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes an e-identity ID of an addressee and encryption option (whether the receiving client encrypts e-document or not).

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — The communication server requests information of an e-identity to the TTP identity directory using the information about the e-identity for which a search was requested by the communication client (call A06 'inquire e-identity information of receiving client' API).

  — Once the network information about the receiving client's agent, receiving server, is received from the TTP identity directory, based on this information an encryption key that will be used in the encryption of the e-document is requested of the receiving server. (call B09 'inquire encrypt information of receiving client' API). If the receiving client does not select an encryption option within a request message, the communication server will omit this stage.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes network information of receiving server acting as proxy of receiving client.

    — If the e-identity ID is not registered in the TTP identity directory, or it is currently unavailable, then the response message will include a failure signal, error code and error description as a result.

  — Attachment in 2nd ~ Nth content part

    — Nothing

C.2.4 B04 'request e-document transmission' API

This is an API called by communication client when the communication client requests the transmission of e-documents to the communication server acting as a proxy of the communication client. At this time, the communication server that receives a request will perform the role of transmitting the e-document, so it will be named a transmitting server.

When the communication client requests a transmission message of the communication server, two different options exist. The first is, prior to the communication client requesting transmission, it acquires the necessary information by searching for information on receiving client and then requests a transmission to the transmitting server based on this information. After that the transmitting server performs the transmission. This step is performed according to the procedure of 5.2.4.2. Another option is that the communication client to request a transmission of e-documents including confirmation of receiving client and encryption of e-documents to the communication server in its entirety. In the case the communication server performs the transmission according to the procedure of 5.2.4.3.

a) Related process

— PR4-1 (e-document transmitting process – basic type) (see 5.2.4.2) and PR4-2 (e-document transmitting process – server delegation type) (see 5.2.4.3).

b) Description of API

— Requester

  — The communication client is a requester that requests B04 'request e-document transmission' API to the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes request option type ('basic type' or 'server delegation type'), e-identity ID of transmitting client and receiving client, perusal confirmation option (whether transmitting client requests to receiving client for perusal confirmation or not) and the other data (e.g. title of transmitting message, description).

    — And if request option type is 'basic type' then it includes network information and security information of receiving server.

    — Else if request option type is 'server delegation type' then it includes encryption option for e-documents.

  — Attachment in 2nd ~ Nth content part

    — Starting from the 2nd content part, each of the e-documents are added.

— Internal process

  — After having extracted the request option type from the message received from the transmitting client, the transmitting server would implement the following procedure.

  — If the request option type is 'basic type' then the transmitting server

    — Extracts the network and security information necessary for connecting to the receiving server and e-documents that will be transmitted to the receiving client from the received message.

    — Following the agreed upon protocol among TCP members, the content transmitted to the receiving client is packaged, and then using the network and security information of the receiving server transmitting server calls B10 'transmit e-document' API provided by the receiving server.

    — Once the transmitting server receives a response message as a result of the API call from the receiving server, it saves the response message.

    — If the response message states 'receipt notification ACK', then the transmitting server would generate a TCE based on the information of the transmission, (such as an e-identity ID of transmitting client and receiving client, transmitting time, receiving time, title of transmitting message, hash values of transmitted contents).

  — Else if the request option type is 'server delegation type', then the transmitting server:

    Extracts the e-identity ID of receiving client in the message received from the transmitting client.

    — In order for the transmitting server to acquire the necessary information (i.e. information about the network and security of the receiving server that acts as receiving client's agent)

when transmitting the e-document to the receiving client, it calls the 'retrieve e-identity information' API provided by the TTP identity director.

- If there is a "yes" for the encryption option for e-documents, using the receiving server's network and security information, the transmitting server calls the 'inquiry information of receiving client' API provided by the receiving server, and acquires an encryption key as a response of the API. And then the transmitting server encrypts the e-documents using the encryption key (optional step).
- Following the agreed upon protocol among TCP members the transmitting server packages the message information and e-documents to be transmitted to the receiving client, and calls B10 'transmit e-document' API provided by the receiving server.
- Once the transmitting server receives a response message as a result of the API call from the receiving server, it saves the response message.
- If the response message states 'receipt notification ACK,' then the transmitting server would generate a TCE based on the information of the transmission, (such as an e-identity ID of transmitting client and receiving client, transmitting time, receiving time, title of transmitting message, hash values of transmitted contents).

- Structure of message replied by responder
  - Response message in message body of 1st content part
    - Includes success or failure as a result of internal process. (If the transmitting server receives a 'receipt notification ACK' as the response of B10 'transmit e-document' API, then the message will include a success value, if not, then the message will include a failure value. And the transmitting server sends the response message to the transmitting client.)
    - If the result is failure then response message includes error code, error description, etc.
  - Attachment in 2nd ~ Nth content part
    - Nothing.

C.2.5 B05 'inquire list of transmission/reception messages' API

The communication client can request a list of messages received or sent by the communication server. This API is used for this request. The communication server received the request queries the message list according to the query conditions and transmits the list to the communication client. The transmitting or receiving time of the communication server(s) should be stored. It should be agreed in the SLA (service level agreement) made between the communication server and the client.

a) Related process

- No related process.

b) Description of API

— Requester

  — The communication client is a requester that requests B05 'inquire list of transmission/ reception messages' API to the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes query condition (e.g. whether a message is transmitted one or received one, transmitting time, addressee of message, title of message).

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — The communication server searches for messages relevant to the query condition among the messages where the requester's e-identity is the originator or the addressee.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes basic information of messages retrieved (e.g. unique ID of message, e-identity ID of the originator and the addressee, transmitting time, receiving time, title of message).

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.2.6 B06 'inquire information of message and e-documents' API

This is an API called where the communication client requests the communication server of all information of a message (such as title, transmitting server, e-identities of an originator and an addressee, transmitting and receiving time) and e-documents in the message. Usually, after the communication client calls the B05 'inquire list of transmission/reception messages' API to acquire a unique ID of message, the communication client calls this API to get e-documents that are included in messages having the unique ID.

a) Related process

— No related process.

b) Description of API

— Requester

  — The communication client is a requester that requests B06 'inquire of message and e-documents' API to the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes unique ID of message.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — The communication server searches the relevant message using a unique ID of message and checks that the requester has read authority of the message.

  — If the requester has read authority, then the communication server gets all information of the message and e-documents from the message.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process (if communication server reads the message matched with the unique ID successfully, then the result is a success, if not, a failure).

    — If the result is success, then the response message includes all information of message (such as transmission and reception time, e-identity of the originator and the addressee, title of message) and all e-documents contained in message.

    — Else if the result is failure, then the response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.2.7 B07 'request message deletion' API

This API is used when the communication client requests that the communication server deletes a certain message. Usually, after the communication client calls the B05 'inquire list of transmission/ reception messages' API to acquire a unique ID of message, the communication client calls this API to delete the messages having the unique ID.

a) Related process

— No related process.

b) Description of API

— Requester

  — The communication client is a requester that requests B07 'request message deletion' API to the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes unique ID of message.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — The communication server searches the relevant message using a unique ID of message and checks that the requester has delete authority of the message.

  — If the requester has deleted authority, then the communication server deletes this message.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process (if a message matched with the unique ID was deleted successfully, then the result is a success, if not, a failure).

    — If the result is success, then the response message Includes all information of message (such as transmission and reception time, the e-identities of the originator and the addressee, title of message) and all e-documents contained in message.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.2.8 B08 'request report of spam message' API

This API is used at a time when the communication client requests to the communication server reporting a received message as a spam message. Having received this request, the communication server will report this message to the TTP identity directory as a spam message.

a) Related process

— PR8 (spam message handling process) (see 5.2.8).

b) Description of API

— Requester

  — The communication client is a requester that requests B08 'request report of spam message' API to the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes unique ID of message and additional information (e.g. reason that the message is spam).

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — The communication server inserts the received spam report into the receipt list for spam reports to transmit it to the TTP identity directory.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process (if a message matched with the unique ID exists and the requester has the authority to report it as a spam message, then the result is a success, if not, a failure).

    — If the result is failure, then the response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.2.9 B09 'inquire encrypt information of receiving client' API

Prior to the communication server transmitting e-documents, to obtain information about the receiving client, a transmitting client requests information to the receiving client's agent, which is the receiving server. This API is called for this request. This API usually is used for acquiring the receiving client's encryption key, therefore in cases where the e-documents are not required to be encrypted, this step can be omitted.

a) Related process

— PR4-1 (e-document transmitting process - basic type) (see 5.2.4.2) and PR4-2 (e-document transmitting process – server delegation type) (see 5.2.4.3).

b) Description of API

— Requester

  — The transmitting communication server is a requester that requests the receiving communication server, which acts as the receiving clients' agent, for B09 'inquire encrypt information of receiving client'.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes the e-identity ID of the receiving client to be retrieved.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — After extracting the e-identity ID from the request message, the communication server performs a search for user matched with the e-identity ID among users whom the communication server has concluded an SLA with.

  — If there is a user matched with the e-identity ID among users, then the communication server retrieves information involving this user including encryption keys.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of the internal process (if the communication server searched the user matched with the e-identity successfully, then the result is a success; if not, a failure).

    — If the result is success, then the response message includes information about the user (e.g. encryption key of the e-identity, and the other information such as name, contact point information; but according to the authority of the requester and TCP's information protection policy, some information of receiving client deemed as privacy would be restricted that the requester gets it).

    — Else if the result is failure, then the response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.2.10 B10 'transmit e-document' API

Once a transmitting server received a request for transmitting e-documents from a transmitting client via B04 'request e-document transmission' API and then sends the e-documents to the receiving server. This API is called for this action by the transmitting server. All communication servers shall provide this API to act as receiving server that receives e-documents.

a) Related process

— PR4-1 (e-document transmitting process – basic type) (see 5.2.4.2) and PR4-2 (e-document transmitting process – server delegation type) (see 5.2.4.3).

b) Description of API

— Requester

  — The communication server that intends to transmit e-documents is a requester that calls the B10 'transmit e-document' API provided by relying communication server playing the role of receiving server. At this time the communication server transmitting e-documents is named by the transmitting server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes request option type (‘process type 1’ or ‘process type 2’), the e-identity ID of the transmitting client and the receiving client, perusal confirmation option (whether transmitting client requests to receiving client for perusal confirmation or not) and the other data (e.g. title of transmitting message, description).

  — Attachment in 2nd ~ Nth content part

    — Starting from the 2nd content part, each one of the e-documents is added.

— Internal process

  — After verifying the validity (verification to verify that the transmitting server is a communication server included in the whitelist and that the transmitting client is not included in the blacklist, and to verify that the receiving client is a proper user that has concluded the service use agreement with the receiving server) of the received message, if the verification results are valid then the receiving server extracts the attached e-documents and the information of the message from the request message.

  — After saving the received message into the receiving client's inbox, the receiving server generates an ACK signal for receipt confirmation such as receipt notification ACK or perusal notification ACK (see 6.2.2.5).

  — The receiving server should record the yes/no status of the perusal confirmation option extracted from the received message. If it is a yes, then after the receiving client reads the message received, the receiving server should perform the perusal confirmation process (see 5.2.5).

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process.

    — If the result is success then response message includes ACK signal for receipt confirmation.

    — Else if the result is failure then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.2.11 B11 ‘deliver confirmation of perusal’ API

Once the receiving server confirms that the message received was read by the receiving client, the receiving server checks that the perusal confirmation options of the message is a "yes." If the transmitting server requests the perusal confirmation option as “yes”, then the receiving server calls this API provided by the transmitting server. Via calling this API, the receiving server sends relevant information of receiving client’s perusal notification ACK to the transmitting server (see 6.2.2.5).

The receiving server determines the time when the message was read as the time that the receiving client received 'success' as a result after calling the B06 'inquire information of message and e-documents' API.

a) Related process

— PR5(perusal confirmation process) (see 5.2.5).

b) Description of API

— Requester

  — The communication client is a requester that requests B11 'deliver confirmation of perusal' API to the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes ACK signal for perusal confirmation.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — The communication server confirms the perusal ACK signal included in a request message received by relying communication server, receiving server. And then the communication server verifies that the perusal ACK signal is matched with the message sent by itself.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process.

    — If the result is failure then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.2.12 B12 'notify result of review of spam message' API

After reviewing the reported message from the communication server via A08 'submit report of spam message' API, the TTP identity directory will determine whether the message is spam or not. Criterions and methods for reviewing are determined by TCP's policy. Once such reviews are completed, the TTP identity directory sends the result of review to the communication server reporting the spam message.

If the spam message was determined not to be a spam, then a report process of the spam message will end at this stage, but if determined to be a spam, the TTP identity directory will add the originator of the spam message to the blacklist and follow the procedure for notifying all communication servers about this within the TCP.

a) Related process

— PR8 (spam message handling process) (see 5.2.8).

b) Description of API

— Requester

  — The TTP identity directory is a requester that requests B12 'notify result of review of a spam message' API to the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes unique message ID reported as a spam message in previous step.

    — Reviews results of the status of the message reported as a spam message.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — The communication server will confirm whether the review result received is matched with the reported message by itself. And the communication server records the review result in its internal system.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of receipt of the submitted message.

    — If the result is failure then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

C.2.13 B13 'notify whitelist' API

The TTP identity directory records and manages information of all communication servers which acts as an agent for reliable communication of e-document in the whitelist. Before each communication server communicates e-documents reliably to each other, the whitelist shall first be distributed to all communication servers who will use it to confirm their counterparts.

To do this, once the communication server is newly registered or deleted or the network information of communication server required for the connection is revised by the A01 'register communication server' API and A03 'manage information of communication server' API, this change would be reflected on the whitelist. After that, TTP identity directory shall notify the changed whitelist to all communication servers using B13 'notify whitelist' API.

a) Related process

— PR1 (communication server registration process) (see 5.2.1).

b) Description of API

— Requester

  — The TTP identity directory is a requester that requests B13 'notify whitelist' API to the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes a latest full whitelist or list of changed (newly registered, deleted or updated) communication servers including its network information.

  — Attachment in $2^{nd} \sim N^{th}$ content part

    — Nothing.

— Internal process

  — The communication server saves the latest full whitelist received or receives the list of changed communication servers and reflects this change in an internally managed whitelist.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of receipt of the submitted message.

    — If the result is failure then response message includes error code, error description, etc.

  — Attachment in $2^{nd} \sim N^{th}$ content part

    — Nothing.

C.2.14 B14 'notify blacklist' API

Once it is concluded that the message reported from the communication server is a spam message, the TTP identity directory will add the e-identity of originator of the spam message to a blacklist. If the TTP identity directory adds or removes an e-identity to or from a blacklist, a notification included the changed blacklist will be sent to all communication servers using B14 'notify blacklist' API.

After receiving the notified blacklist, the communication server shall internally store and manage it. If an e-identity registered in this blacklist sent a message via a communication client, then the receiving server can refuse to receive the message fairly.

a) Related process

— PR8 (spam message handling process) (see 5.2.8).

b) Description of API

— Requester

  — The TTP identity directory is a requester that requests B14 'notify blacklist' API to the communication server.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes a latest full blacklist or list of newly registered or deleted e-identity in the blacklist.

  — Attachment in $2^{nd} \sim N^{th}$ content part

    — Nothing.

— Internal process

  — The communication server saves the latest full blacklist received or receives the list of newly registered or deleted e-identities and reflects this change in an internally managed blacklist.

— Structure of message replied by responder

  — Response message in message body of 1^{st} content part

    — Includes success or failure as a result of receipt of the submitted message.

    — If the result is failure then response message includes error code, error description, etc.

  — Attachment in $2^{nd} \sim N^{th}$ content part

    — Nothing.

C.3 APIs of TCE repository

C.3.1 C01 'store TCE' API

Using this API, a communication server stores a TCE generated as an evidence for trusted e-document communication into TCE repository to preserve it safely and reliably.

After finishing trusted communication, the transmitting server receives a notification ACK for receiving or perusal and based on this signal generates a TCE (see Annex A).

The communication server stores TCE generated in the storage, TCE repository where is a technically sound and reliable place, by calling this API. In doing so, later when a dispute related with a communication arises, a behaviour of communication can be verified based on the stored TCE information.

a) Related process

— PR6 (TCE preservation process) (see 5.2.6).

b) Description of API

— Requester

  — The communication server is a requester that requests a C01 'store TCE' to the TCE repository API.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes TCE.

  — Attachment in $2^{nd} \sim N^{th}$ content part

    — Nothing.

— Internal process

  — The TCE repository verifies whether the structure and digital signature of the TCE in request message are valid, and whether the communication server that requested preservation is a server that is registered on the whitelist or not.

  — If the requester is a legitimate server, and the structure of the TCE is valid, the TCE repository will then save it internally.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process.

    — If the result is failure then response message includes error code, error description, etc.

  — Attachment in $2^{nd} \sim N^{th}$ content part

    — Nothing.

C.3.2 C02 'communication verification' API

This API is called when a communication server or communication client request to verify a behaviour of communication that occurred in the past to TCE repository. The TCE repository searches the TCE using the unique ID of TCE presented by the communication server or client. After that, TCE repository extract information related the communication from the TCE and returns the information to the requester thereby verifying its trusted communication.

a) Related process

— PR7 (communication verification process) (see 5.2.7).

b) Description of API

— Requester

  — The communication server or client is a requester that requests a C02 'communication verification' API to the TCE repository.

— Structure of message sent by requester

  — Request message in message body of 1st content part

    — Includes ID of TCE.

  — Attachment in 2nd ~ Nth content part

    — Nothing.

— Internal process

  — The TCE repository searches a TCE matched with the ID of TCE in request message.

  — If there is a TCE matched with the ID, TCE repository extracts the information related to trusted communication from the TCE.

— Structure of message replied by responder

  — Response message in message body of 1st content part

    — Includes success or failure as a result of internal process (if a TCE matched with the ID exists it is a success, if not, a failure).

    — If the result is success then response message includes information recorded in TCE (such as TCE type whether TCE is for receiving confirmation or perusal confirmation, information of originator and addressee, transmitting time, receiving time, hash values of attached e-documents).

    — Else if the result is failure then response message includes error code, error description, etc.

  — Attachment in 2nd ~ Nth content part

    — 2nd content includes TCE.

Bibliography

[1] IETF RFC 3852, *CMS (Cryptographic Message Syntax)*

[2] ISO 15000-2, *Electronic business eXtensible markup language (ebXML) – Part 2: Message service specification (ebMS)*

[3] ISO 15489-1, *Information and documentation — Records management — Part 1: Concepts and principles*

[4] ISO/TR 15801, *Document management — Electronically stored information — Recommendations for trustworthiness and reliability*

[5] ISO/TS 16175-2, *Information and documentation — Processes and functional requirements for software for managing records — Part 2: Guidance for selecting, designing, implementing and maintaining software for managing records*

[6] ISO 17068, *Information and documentation — Trusted third party repository for digital records*

[7] ISO 31010, *Risk management — Risk assessment techniques*

[8] ISO 9000, *Quality Management Systems — Fundamentals and Vocabulary*

[9] ISO 9735-5, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 5: Security rules for batch EDI (authenticity, integrity and non-repudiation of origin)*

[10] ISO 9735-6, *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1) — Part 6: Secure authentication and acknowledgement message (message type - AUTACK)*

[11] ISO/IEC 27033-1:2015, *Information technology — Security techniques — Network security — Part 1: Overview and concepts*

[12] ISO/IEC 14662, *Information technology — Open-edi reference model*

[13] ITU-T 2008, *X.1152: Secure end-to-end data communication techniques using trusted third party services*

[14] ITU-T 2000, *X.842: Information technology – Security techniques - Guidelines for the use and management of trusted third party services*

[15] UNCEFACT 2014, *Revision of Recommendation 14: Authentication of trade documents*

[16] UNCITRAL 1996, *Model Law on e-Commerce*

[17] UNCITRAL 2001, *Model Law on e-Signature*

[18] UNCITRAL 2007, *United Nations Convention on the Use of Electronic Communications in International Contracts*

Bureau of Indian Standards

BIS is a statutory institution established under the *Bureau of Indian Standards Act*, 2016 to promote harmonious development of the activities of standardization, marking and quality certification of goods and attending to connected matters in the country.

Copyright

Review of Indian Standards

Amendments are issued to standards as the need arises on the basis of comments. Standards are also reviewed periodically; a standard along with amendments is reaffirmed when such review indicates that no changes are needed; if the review indicates that changes are needed, it is taken up for revision. Users of Indian Standards should ascertain that they are in possession of the latest amendments or edition by referring to the website-www.bis.gov.in or www.standardsbis.in.

This Indian Standard has been developed from Doc No.: MSD 05 (23520).

Amendments Issued Since Publication

Amend No.	Date of Issue	Text Affected

BUREAU OF INDIAN STANDARDS

Headquarters:

Manak Bhavan, 9 Bahadur Shah Zafar Marg, New Delhi 110002
Telephones: 2323 0131, 2323 3375, 2323 9402 *Website*: www.bis.gov.in

Regional Offices:		*Telephones*
Central	: 601/A, Konnectus Tower -1, 6th Floor, DMRC Building, Bhavbhuti Marg, New Delhi 110002	2323 7617
Eastern	: 8th Floor, Plot No 7/7 & 7/8, CP Block, Sector V, Salt Lake, Kolkata, West Bengal 700091	2367 0012 2320 9474
Northern	: Plot No. 4-A, Sector 27-B, Madhya Marg, Chandigarh 160019	265 9930
Southern	: C.I.T. Campus, IV Cross Road, Taramani, Chennai 600113	2254 1442 2254 1216
Western	: 5th Floor/MTNL CETTM Technology Street, Hiranandani Gardens, Powai, Mumbai - 400076	283 25838

Branches : AHMEDABAD, BENGALURU, BHOPAL, BHUBANESHWAR, CHANDIGARH, CHENNAI, COIMBATORE, DEHRADUN, DELHI, FARIDABAD, GHAZIABAD, GUWAHATI, HARYANA (CHANDIGARH), HUBLI, HYDERABAD, JAIPUR, JAMMU, JAMSHEDPUR, KOCHI, KOLKATA, LUCKNOW, MADURAI, MUMBAI, NAGPUR, NOIDA, PARWANOO, PATNA, PUNE, RAIPUR, RAJKOT, SURAT, VIJAYAWADA.

Published by BIS, New Delhi