D8.4

Conformance and Interoperability Testing Result Report (2)

Document Identification	
Date	25.02.2019
Status	Final
Version	1.0

Related WP	WP3, WP4, WP5, WP6	**Related Deliverable(s)**	D2.3, D2.14, D3.3, D3.4, D4.3, D4.4, D5.2, D8.3
Lead Authors	TUBITAK	**Dissemination Level**	PU
Lead Participants	TUBITAK	**Contributors**	TUBITAK
Reviewers	USTUTT, ATOS		

This document is issued within the frame and for the purpose of the LIGHT*est* project. LIGHT*est* has received funding from the European Union's Horizon 2020 research and innovation programme under G.A. No 700321.

1. Executive Summary

This document is the deliverable D8.4 “Conformance and Interoperability Testing Result Report (2)” of the project “Lightweight Infrastructure for Global Heterogeneous Trust management in support of an open Ecosystem of Stakeholders and Trust schemes” (LIGHT*est*, project nr. 700321) with the objective to create a global cross domain trust infrastructure that renders it transparent and easy for verifiers to evaluate electronic transactions.

This document presents the second iteration of the “Conformance and Interoperability Testing” and is built upon D8.3 Conformance and Interoperability Testing Result Report (1) where test assertions and normative specifications are derived from design and use case documentations. In this document, test cases that are derived from TSPA, TTA, and DP assertions that are given in D8.3. Results of the testing will be provided in the last iteration of the report.

2. Document Information

2.1 Contributors

Name	Partner
Asiye BOZKURT	TUBITAK
Neslihan KIZILBEY	TUBITAK
Sedat ÇİFTÇİ	TUBITAK
Burçin BOZKURT GÜNAY	TUBITAK
Berkay TOPÇU	TUBITAK

2.2 History

Version	Date	Author	Changes
0.0	04/12/2018	Burçin BOZKURT GUNAY	First Draft, TOC
0.1	14/01/2019	TUBITAK	Initial Version
0.2	12/02/2019	TUBITAK	Test Case Updates
0.3	11/02/2019	TUBITAK	General review and updates
1.0	25/02/2019	TUBITAK	Final Version

3. Table of Contents

1. Executive Summary 2

2. Document Information 3
2.1 Contributors 3
2.2 History 3

3. Table of Contents 4
3.1 Table of Figures 6
3.2 Table of Tables 6
3.3 Table of Acronyms 6

4. Scope of the deliverable 8
4.1 Overview 8
4.2 Scope 8

5. Testing Architecture 9

6. Testing Methodology 11
6.1 Test Case Generation Guideline 11

7. Conformance and Interoperability Test Assertions and Test Cases 13
7.1 TSPA 13
7.1.1 TSPA Conformance Clauses 13
7.1.2 TSPA Normative Statements 13
7.1.3 TSPA Test Assertions 13
7.1.4 TSPA Test Scenario 16
7.1.5 TSPA Test Cases 18
7.1.5.1 TSPA Test Case List 18
7.1.5.2 TSPA Test Case Details 19
7.2 TTA 38
7.2.1 TTA Conformance Clauses 38
7.2.2 TTA Normative Sources 38
7.2.3 TTA Test Assertions 38
7.2.4 TTA Test Scenario 42
7.2.4.1 Discovery of the translation list for a boolean trust scheme 42
7.2.4.2 Discovery of the translation list for an ordinal trust scheme 43
7.2.4.3 Discovery of the translation list for a tuple trust scheme 44
7.2.5 TTA Test Cases 45
7.2.5.1 TTA Test Case List 45
7.2.5.2 TTA Test Case Details 45
7.3 DP 67

7.3.1 DP Conformance Clauses 67
7.3.2 DP Normative Statements 68
7.3.3 DP Test Assertions 68
7.3.4 DP Test Scenario 68
7.3.4.1 Test Scenario for DP (publication & download services): 68
7.3.4.2 Test Scenario for DP (revocation & search services) 69
7.3.5 DP Test Cases 70
7.3.5.1 DP Test Case List 70
7.3.5.2 DP Test Case Details 70

8. References 78

9. Project Description 79

3.1 Table of Figures

Figure 1 Minder Testbed Applied Architecture 9
Figure 2 General Anatomy of Test Model 11

3.2 Table of Tables

Table 1 List of TSPA test cases 18
Table 2 List of TTA test cases 45
Table 3 List of DP test cases 70

3.3 Table of Acronyms

API	Application Program Interface
ATV	Automatic Trust Verifier
CC	Conformance Clause
DNS	Domain Name System
DNSSEC	Domain Name System SECurity extensions
DP	Delegation Publisher
eIDAS	Electronic Identification, Authentication and trust (Services)
eT	Electronic transaction
FR	Functional Requirement
HTTP(S)	Hypertext Transfer Protocol (Secure)
ISTQB	International Software Testing Qualifications Board
MTDL	Minder Test Definition Language
M1	Minder END User ATV Adapter
M2	Minder ATV Adapter
NS	Normative Statement
OASIS	Advancing Open standards for information society
PDF	Portable Document Format
RA	Reference Architecture
PTR	Pointer
REST	Representational State Transfer (service)
RR	Resource Record
S/MIME	Secure/Multipurpose Internet Mail Extensions
SUT	System Under Test
TA	Test Assertion
TP	Trust Policy
TPL	Trust Policy Language
TSLTS	Trust Service Status List Technical Specification
TSL	Trust Service Status List
TSP	Trust Service Provider
TSPA	Trust Scheme Publication Authority
TTA	Trust Translation Authority
URI	Uniform Resource Identifier

XML	Extensible Markup Language

4. Scope of the deliverable

4.1 Overview

The overall focus of the LIGHT*est* project is to develop a lightweight trust infrastructure providing parties of electronic transactions with automatic validation of trust based on their individual trust policies. By using an existing infrastructure of the global Domain Name System (DNS) for publication, querying, and cross-jurisdiction translation of information relevant to make such decisions, including levels of assurance, LIGHT*est* aims to enable the use of truly "global trust lists". With this approach LIGHT*est* will basically provide an infrastructure to realize the most important principles and driving factors of eIDAS on a global level.

Conformance testing, also known as compliance testing, is a methodology used in software engineering to ensure that a product, process, computer program or system meets a defined set of standards. In this task, we will test outputs of other WPs in order to see whether they conform to the proposed specifications and standards. Interoperability testing, on the other hand, verifies whether all the systems exchange and use information properly, interpret the exchanged information meaningfully, and multiple entities work together in a harmonious way.

This deliverable is structured as follows. Section 1 presents the executive summary. Section 2 basically includes document information and Section 3 gives the table of contents. Section 4 presents an overview of WP8 and scope of this deliverable. Section 5 summarizes the testing architecture with Minder Test Manager inclusion and Section 6 revisits the testing methodology given in D8.3. Section 6 presents the test cases for TSPA, TTA, and DP derived from conformance and interoperability test assertions of TSPA, TTA and DP. Finally, Section 7 concludes the deliverable.

4.2 Scope

Within the course of the LIGHTest project, conformance and interoperability testing for the software components developed in WP3, 4, 5, and 6 will be carried out by using Minder Testbed and the results will be reported periodically. D8.4 Conformance and interoperability testing report (2) is the second report of this series and focuses on defining test cases to perform conformance and interoperability testing of the LIGHTest components.

The main contents of this deliverable include test cases that are derived from normative statements and test assertions given in D8.3 according to the applied Minder Test Assertion Model. Test assertions and the testing environment architecture have been updated to enhance the execution of the test cases and reporting.

Section 5 content is updated in this deliverable to include the implementation of ATV-Minder and Minder Test Manager components for testing. Section 6 content is updated to include the derivation methodology of test cases from test assertions. Section 7 includes the test scenario and test cases for TTA, TSPA and DP components.

Results of the test executions will be provided in the last iteration of the report.

5. Testing Architecture

The LIGHTest reference architecture (RA) is already given in Section Testing Architecture in D8.3. There has been an update on the Minder Testbed Applied Architecture in the sense that Minder Test Manager application and Minder-ATV are implemented for testing purposes.

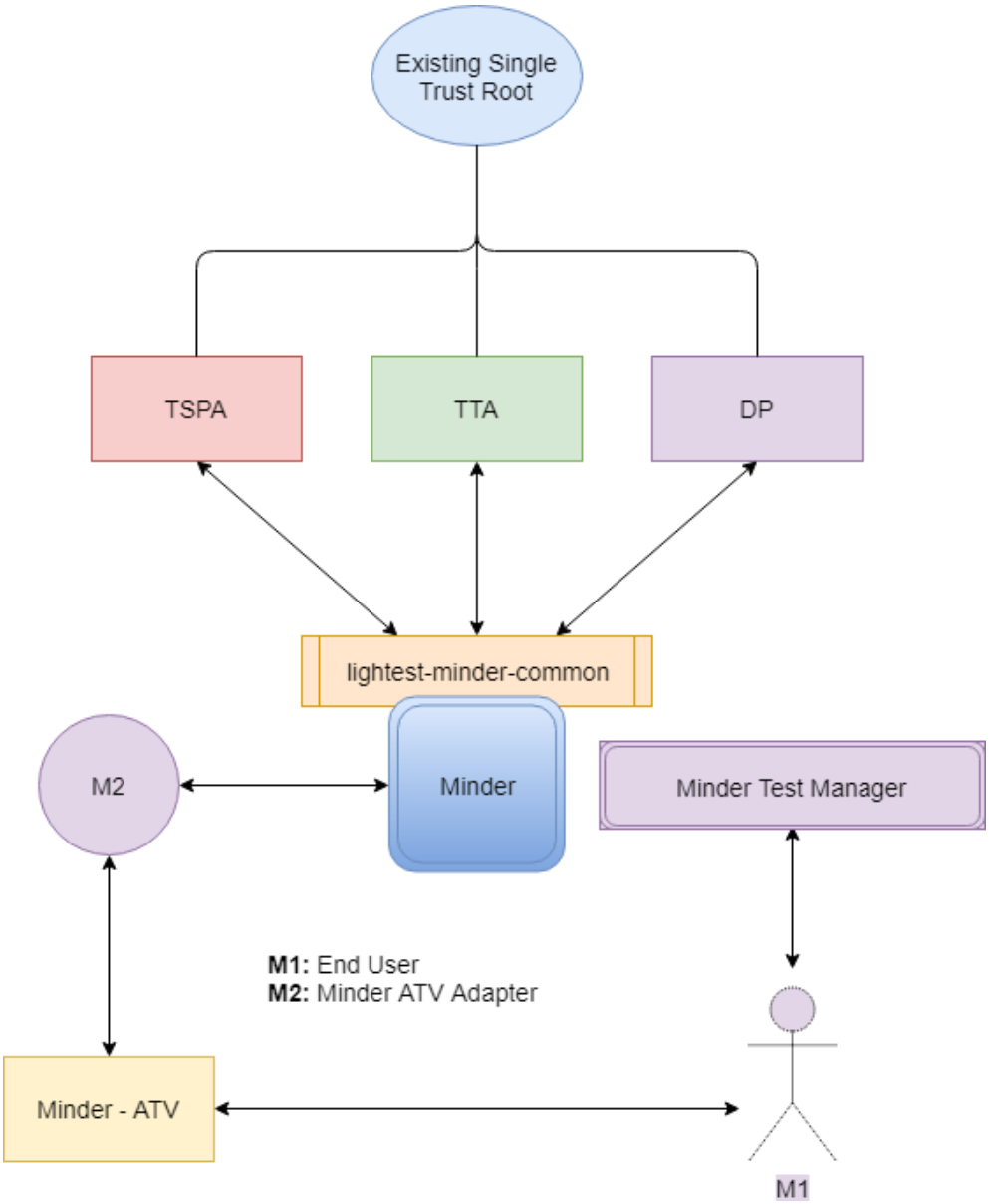

Figure 1 Minder Testbed Applied Architecture

The communication between Minder and the TSPA, and TTA, can be handled directly via Minder's own DNS Client component, likewise, the communication between Minder and the DP can be handled via Minder's HTTP component eliminating the need for adapters for all these components.

Automatic Trust Verifier (Minder – ATV) component has been re-implemented/developed and included in the testing architecture for testing purposes to manage the scenarios that include the usage of REST services provided by TSPA, TTA and DP components.

In addition, features that support the management of test cases/suits and test executions have been developed within Minder Testbed architecture. The new feature is called as "Minder Test Manager" and the source codes are included in the Minder Testbed sources. The "Minder Test Manager" also uses the same DNS Client and HTTP Client components to gain access to the APIs provided by TSPA, TTA and DP components.

6. Testing Methodology

The main testing methodology is already described in D8.3 Section 6.

Figure 2 depicts the general anatomy of a OASIS Test Model Case, where Test cases are derived from Test assertions in form of Test Suites.

Test Suite/Case management feature is supported by Minder Test Manager.

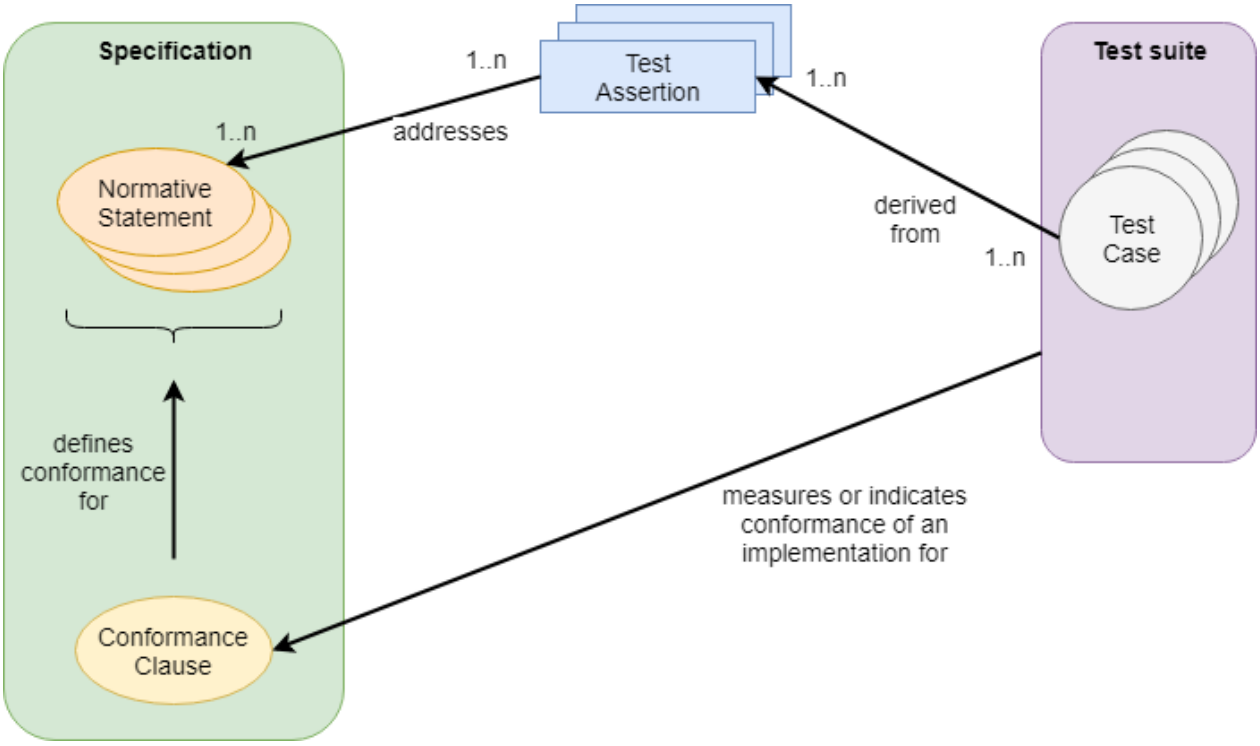

Figure 2 General Anatomy of Test Model

6.1 Test Case Generation Guideline

Following the same methodology in D8.3 Section 6, test cases are derived from test assertions. The details about the test case derivation methodology is given in *Appendix A. Deriving a Test Case from a Test Assertion* given in (http://docs.oasis-open.org/tag/guidelines/v1.0/guidelines-v1.0.pdf).

As a summary, conditions to derive a test case from a test assertion are given as follows:

- When a Target instance is not qualified for a Test Assertion, a Test Case derived from this Test Assertion does not indicate whether the Target instance fulfills or not the Normative Statement addressed by the Test Assertion.

- When a Target instance is qualified for a Test Assertion and satisfies the Test Assertion Predicate, a Test Case derived from this Test Assertion either indicates that the Target instance fulfills the Normative Statement addressed by the test assertion, or does not indicate anything.

- When a Target instance is qualified for a Test Assertion and does not satisfy the Test Assertion Predicate, a Test Case derived from this Test Assertion either indicates that the Target instance does not fulfill the Normative Statement addressed by the test assertion, or does not indicate anything.

7. Conformance and Interoperability Test Assertions and Test Cases

This section lists the second round of test assertions for TSPA, TTA, and DP, and first round of test cases derived from test assertions TSPA, TTA, and DP for respectively. Test assertions that were extracted from the respective Conceptual Framework and Design deliverables of these components are already given in D8.3. This section includes the updated version of assertions. Also, this section includes the initial version of test cases that will be used for test executions.

Following the methodology described in the previous section, we first analyze the normative sources together with their references, and the test assertions to elicitate test cases for conformance to these specifications.

For traceability purposes, the identifiers of the test was done using the following convention:
TC_LightestComponentName(TSPA,TTA,DP)_TestCaseID

7.1 TSPA

7.1.1 TSPA Conformance Clauses

The conformance clauses are already provided in D8.3 TSPA Conformance Clauses section.

7.1.2 TSPA Normative Statements

The normative statements are already provided in D8.3 TSPA Normative Statements section. The only updated normative statement is given below:

NS ID	NS_TSPA_8
Reference	D3.3 Section 6.1
Description	A TSPA is composed of a HTTP Server (Trust Scheme Provider) that contains: a. Signed Trust Lists b. Tuple-based (ordinal and Boolean included) representations of Trust Schemes, provided as pointer from Trust List and as additional information in the Trust List c. deleteService, publishService for trust scheme and trust list, getTrustSchemeService is available

7.1.3 TSPA Test Assertions

The test assertions are updated and given in this section.

TA ID	TA_TSPA_1
Normative Source	NS_ TSPA_1
Target	TSPA
Prerequisite	
Prescription Level	Mandatory

Predicate	IP address of the TSPA DNS server exists and can be listed on the configurations and is already set on the TCP/IP Properties (DNS Server Address settings)

TA ID	TA_TSPA_2
Normative Source	NS_ TSPA_1
Target	TSPA
Prerequisite	The TSPA DNS Name Server is up and running and TSPA contains published trust scheme membership declarations.
Prescription Level	Mandatory
Predicate	The RR responses to the TSPA scheme membership query (IssuerName, SchemeName, CertificateConstraints) are signed by a valid Zone Key.

TA ID	TA_TSPA_3
Normative Source	NS_TSPA_6, NS_TSPA_7
Target	TSPA
Prerequisite	The TSPA DNS Name Server is up and running and contains published trust scheme membership declarations in the form of PTR Records. The ATV has issued an IssuerName query to the TSPA.
Prescription Level	Mandatory
Predicate	The RR response to the IssuerName query is a PTR Record and its DNSSEC validation is successful.

TA ID	TA_TSPA_4
Normative Source	NS_TSPA_6, NS_TSPA_7, NS_TSPA_16
Target	TSPA
Prerequisite	The TSPA DNS Name Server is up and running and contains published scheme locations declarations in the form of URI Records. The ATV has issued a SchemeNameLocation query to the TSPA.
Prescription Level	Mandatory
Predicate	The RR response to the SchemeNameLocation query is a URI Record and its DNSSEC validation is successful.

TA ID	TA_TSPA_5
Normative Source	NS_TSPA_6, NS_TSPA_7, NS_TSPA_17
Target	TSPA
Prerequisite	The TSPA DNS Name Server is up and running. The ATV has issued an IssuerName_SchemeNameAssociation query to the TSPA.
Prescription Level	Mandatory

Predicate	The RR response to the IssuerName_SchemeNameAssociation query is a signed association and its signature validation is successful.

TA ID	TA_TSPA_6
Normative Source	NS_TSPA_13, TA_TSPA_3
Target	TSPA
Prerequisite	The TSPA DNS Name Server is up and running and contains published trust scheme membership declarations in the form of PTR Records. The ATV has issued an IssuerName query to the TSPA.
Prescription Level	Mandatory
Predicate	The received DNS query is of the form _scheme._trust.IssuerDomainName IN PTR

TA ID	TA_TSPA_7
Normative Source	NS_TSPA_15, NS_TSPA_9, TA_TSPA_4
Target	TSPA
Prerequisite	The TSPA DNS Name Server is up and running and contains published trust scheme membership declarations in the form of PTR Records. The ATV has issued an SchemeNameLocation query to the TSPA.
Prescription Level	Mandatory
Predicate	The received DNS query is of the form _scheme._trust.SchemeNameDomainName IN URI

TA ID	TA_TSPA_8
Normative Source	NS_TSPA_18, TA_TSPA_8
Target	TSPA
Prerequisite	The TSPA DNS Name Server is up and running and contains published trust scheme membership declarations in the form of PTR Records. The ATV has issued a CertificateConstraints query to the TSPA.
Prescription Level	Mandatory
Predicate	The received DNS query is of the form scheme._trust. SchemeNameDomainName IN SMIMEA

TA ID	TA_TSPA_09
Normative Source	NS_TSPA_14, NS_TSPA_12, NS_TSPA_10
Target	TSPA
Prerequisite	The TSPA DNS Name Server is up and running and contains published trust scheme membership declarations in the form of PTR Records. The ATV has issued an IssuerName query to the TSPA.

Prescription Level	Mandatory
Predicate	The RR response to the IssuerName query is a PTR Record containing the domain name of the SchemeName if the queried trust scheme is Boolean.

TA ID	TA_TSPA_10
Normative Source	NS_TSPA_14, NS_TSPA_12, NS_TSPA_10
Target	TSPA
Prerequisite	The TSPA DNS Name Server is up and running and contains published trust scheme membership declarations in the form of PTR Records. The ATV has issued an IssuerName query to the TSPA.
Prescription Level	Mandatory
Predicate	The RR response to the IssuerName query is a PTR Record containing levelName.domainName of the SchemeName if the queried trust scheme is Ordinal.

TA ID	TA_TSPA_11
Normative Source	NS_TSPA_8
Target	TSPA
Prerequisite	The TSPA DNS Name Server is up and running and contains published trust scheme membership declarations in the form of PTR Records. The ATV has issued an IssuerName query to the TSPA.
Prescription Level	Mandatory
Predicate	A delete service should be provided by TSPA

TA ID	TA_TSPA_12
Normative Source	NS_TSPA_6, NS_TSPA_7
Target	TSPA
Prerequisite	The TSPA DNS Name Server is up and running and contains published trust list declarations in the form of signed trust lists. The ATV has issued an SchemeNameTuples query to the TSPA.
Prescription Level	Mandatory
Predicate	The RR response to the SchemeNameTuples query is a set of tuples retrieved from the pointer of the respective trust list entry.

7.1.4 TSPA Test Scenario

Electronic Signature Law of Turkey:

Turkey does not have a public trust list published under any domain presently. Therefore, the process described here is from an educated guess on how it should be accessed if it was

published on a domain (turkey.lightest.nlnetlabs.nl) We use the eIDAS format as a template for Turkey also.

The trust verification for a certificate issued by an eIDAS qualified trust service is shown on tests. The name of the trust service is “eIDAS electronic signature trust service”. It is assumed that there is a Root CA and 2 issuers under this CA (issuera and issuerb) It is assumed that the electronic transaction is simply a signed document containing an invoice prepared according to “urn:oasis:names:specification:ubl:schema:xsd:Invoice-2” schema.

It is signed with a certificate issued by the Turkey Trust Provider (A fictional trust provider), which is a qualified trust service provider for Turkey (signerA). Either the certificate contains the Issuer Alternative Name extension with a domain name value of **turkey.lightest.nlnetlabs.nl** or the issuer certificate (issuerA) used for signing the certificate contains the Subject Alternative Name with that domain name value.

In order to discover the trust scheme(s) this trust service is a member, the verifier will perform a DNS query for PTR records.

This indicates to the verifier that Turkey Trust Provider claims a membership with a trust scheme identified as ***eidas.kamusm.gov.tr-example*** (which is the fictional domain used by Turkey for their trusted list owned by Kamu SM in Turkey).

The verifier will have to discover the trust list for that via another DNS query. It should download that list and see if the issuer certificate (**signerA**) from the electronic transaction appears on that list.

The trust list is signed with “**Turkey Trust Provider CA TR”.** The verifier will check whether the “**Turkey Trust Provider CA TR”** used for signing the trusted list is valid with SMIMEA record.

Assuming that all of this is checked out, the verifier now knows that the electronic transaction was signed by a certificate issued through the trust scheme: *eidas.kamusm.gov.tr-example.*

The published Trust Service Status List for Turkey includes the trust scheme memberships of the *eidas.kamusm.gov.tr-example... Scheme.* It includes the issuers that claims the membership for the *eidas.kamusm.gov.tr-example.*

From the signature part of *http://www.mindertestbed.org:8081/trust/TSL-XML.xml,* certificate is extracted, <ds:Signature> part is taken into consideration.

There will also be an invalid trust list for the *eidas.kamusm.gov.tr-example* trust scheme with

http://www.mindertestbed.org:8081/trust/InvalidTSL-XML.xml

An invalid SMIMEA record will be defined on the DNS to check an invalid trust list validation.

A trust scheme under turkey.lightest.nlnetlabs.nl domain is to be used for *boolean.eidas.kamusm.gov.tr-example*

A trust scheme under turkey.lightest.nlnetlabs.nl domain is to be used for *ordinallevelname.eidas.kamusm.gov.tr-example*

A trust scheme under turkey.lightest.nlnetlabs.nl domain is to be used for *tuplesnameandbirth.eidas.kamusm.gov.tr-example*

A trust list that is to be used by deleteService is to be used with *http://www.mindertestbed.org:8081/trust/TobedeletedTSL-XML.xml*

A trust scheme under turkey.lightest.nlnetlabs.nl domain is to be used for verifying deleting service: *tobedeleted.eidas.kamusm.gov.tr-example*

7.1.5 TSPA Test Cases

This section includes the list of test cases and test case details.

7.1.5.1 TSPA Test Case List

Table 1 List of TSPA test cases

ID	Purpose
TC_TSPA_1	Check if TSPA-DNS server exists and is configured to be used by TSPA.
TC_TSPA_2	Verify that RR responses are signed by a valid Zone Key of DNS
TC_TSPA_3	Verify that RR response to the issuername is a PTR record and DNSSec validation is successfull
TC_TSPA_4	Verify that RR response to the issuername is a URI record and DNSSec validation is successfull
TC_TSPA_5	Verify that trust list,pointed on the URI RR record, signature validation is successful
TC_TSPA_6	Verify that an invalid trust list,pointed on the URI RR record, signature validation fails
TC_TSPA_7	Verify that received DNS query is of the form given in TA_TSPA_6
TC_TSPA_8	Verify that trust scheme and trust lists can be successfully retrieved from TSPA
TC_TSPA_9	Verify that a trust list for the given trust scheme can be published successfully
TC_TSPA_10	Verify that trust list published on TSPA validation fails in case of invalid certificate constraints

TC_TSPA_11	Verify that a Boolean trust scheme can be successfully defined and published on TSPA
TC_TSPA_12	Verify that an ordinal trust scheme can be successfully defined and published on TSPA
TC_TSPA_13	Verify that trust schemes on TSPA can be successfully retrieved
TC_TSPA_14	Verify that received TSPA works in synchronization with DNS entries
TC_TSPA_15	Verify that delete service of TSPA for scheme name works successfuly
TC_TSPA_16	Verify that delete service of TSPA for trust-list works successfully
TC_TSPA_17	Verify that delete service of TSPA for trust-list works properly
TC_TSPA_18	Verify that delete service of TSPA for a trust scheme works in synchronization with DNS records.
TC_TSPA_19	Verify that delete service of TSPA for trust-list works in synchronization with DNS records.
TC_TSPA_20	Verify that tuple schemes can be defined and published on TSPA successfully.

7.1.5.2 TSPA Test Case Details

ID	**TC_TSPA_1**	
Assertion(s)	TA_TSPA_1	
Test Purpose	Check if TSPA-DNS server exists and is configured to be used by TSPA.	
Pre-Test Conditions	TSPA should already be deployed DNS deployment should be available	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig lightest.nlnetlabs.nl*	; <<>> DiG 9.10.6 <<>> lightest.nlnetlabs.nl ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8688 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;lightest.nlnetlabs.nl. IN A ;; ANSWER SECTION: lightest.nlnetlabs.nl. 3600 IN A 185.49.141.61
2	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/scheme with eidas.kamusm.gov.tr-example* scheme name	The service should return HTTP 200 OK

3	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme turkey.lightest.nlnetlabs.nl PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_scheme._trust.turkey.lightest.nlnetlabs.nl. IN PTR ;; ANSWER SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl. 3600 IN PTR _scheme._trust.eidas.kamusm.gov.tr-example

ID	**TC_TSPA_2**	
Assertion(s)	TA_TSPA_2	
Test Purpose	Verify that RR responses are signed by a valid Zone Key of DNS	
Pre-Test Conditions	TSPA DNS is up and running with DANE protocol enabled in the configured DNS	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig lightest.nlnetlabs.nl +noall +comments*	; <<>> DiG 9.10.6 <<>> lightest.nlnetlabs.nl +noall +comments ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10833 ;; flags: qr rd ra aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

ID	**TC_TSPA_3**	
Assertion(s)	TA_TSPA_3	
Test Purpose	Verify that RR response to the issuername is a PTR record and DNSSec validation is successfull	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with *eidas.kamusm.gov.tr-example* should be published *with turkey.lightest.nlnetlabs.nl* domain name in TSPA	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: dig _scheme._trust. *turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme turkey.lightest.nlnetlabs.nl PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761

		;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_scheme._trust.turkey.lightest.nlnetlabs.nl. IN PTR ;; ANSWER SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl. 3600 IN PTR 1 1_scheme._trust.eidas.kamusm.gov.tr-example

ID	**TC_TSPA_4**	
Assertion(s)	TA_TSPA_4	
Test Purpose	Verify that RR response to the issuername is a URI record and DNSSec validation is successfull	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with *eidas.kamusm.gov.tr-example* should be published with *turkey.lightest.nlnetlabs.nl* domain name in TSPA	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list with http://www.mindertestbed.org:8081/trust/TSL-XML.xml* trust list parameter	The service should return HTTP 200 OK
2	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www. mindertestbed.org:8081/trust/TSL-XML.xml*

ID	**TC_TSPA_5**
Assertion(s)	TA_TSPA_5

Test Purpose	Verify that trust list,pointed on the URI RR record, signature validation is successful	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with *eidas.kamusm.gov.tr-example* should be published *with turkey.lightest.nlnetlabs.nl* domain name in TSPA A valid trust list is defined in TSPA (TC_TSPA_4 should be executed)	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www. mindertestbed.org:8081/trust/TSL-XML.xml*
2	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* with *http://www. mindertestbed.org:8081/trust/TSL-XML.xml* parameter	The trust list should be downloaded and should be opened via an XML editor
3	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return TRUE
4	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example SMIMEA*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* SMIMEA ;; ANSWER SECTION: *_scheme._trust.eidas.kamusm.gov.tr-example... IN SMIMEA (3 0 1 0) with the full certificate*

ID	TC_TSPA_6	
Assertion(s)	TA_TSPA_5	
Test Purpose	Verify that an invalid trust list,pointed on the URI RR record, signature validation fails	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with *eidas.kamusm.gov.tr-example* should be published with *turkey.lightest.nlnetlabs.nl* domain name in TSPA	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list with http://www.mindertestbed.org:8081/trust/Invalid-TSL-XML.xml* trust list parameter	The service should return HTTP 200 OK
2	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www. mindertestbed.org:8081/trust/InvalidTSL-XML.xml*
3	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* with http://www. mindertestbed.org:8081/trust/TSL-XML.xml parameter	The trust list should be downloaded and should be opened via an XML editor
4	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return FALSE
5	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example SMIMEA*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

		;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* SMIMEA ;; ANSWER SECTION: *_scheme._trust.eidas.kamusm.gov.tr-example... IN SMIMEA (3 0 1 0) with the full certificate*
6	Execute Minder-ATV *checkCertificatefromSMIMEA* service to verify the certificate used to sign the trust list	Certificate validation result should NOT be successfull

ID	**TC_TSPA_7**	
Assertion(s)	TA_TSPA_6	
Test Purpose	Verify that received DNS query is of the form given in TA_TSPA_6	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with *eidas.kamusm.gov.tr-example* should be published with *turkey.lightest.nlnetlabs.nl* domain name in TSPA	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl*	The service should return HTTP 200 OK
2	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme turkey.lightest.nlnetlabs.nl PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_scheme._trust.turkey.lightest.nlnetlabs.nl. IN PTR ;; ANSWER SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl. 3600 IN PTR 1 1_scheme._trust.eidas.kamusm.gov.tr-example

ID	**TC_TSPA_8**
Assertion(s)	TA_TSPA_7
Test Purpose	Verify that trust scheme and trust lists can be successfully retrieved from TSPA
Pre-Test Conditions	TSPA should already be deployed and running

	DNS deployment with DNSSec extension should be running A valid trust scheme: with *eidas.kamusm.gov.tr-example* should be published with *turkey.lightest.nlnetlabs.nl* domain name in TSPA	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl*	The service should return HTTP 200 OK with response data including eidas.kamusm.gov.tr-example trust scheme
2	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme turkey.lightest.nlnetlabs.nl PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_scheme._trust.turkey.lightest.nlnetlabs.nl. IN PTR ;; ANSWER SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl. 3600 IN PTR 1 1_scheme._trust.eidas.kamusm.gov.tr-example
3	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list with http://www. mindertestbed.org:8081/trust/TSL-XML.xml* parameter	The service should return HTTP 200 OK
4	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www. mindertestbed.org:8081/trust/TSL-XML.xml*

ID	TC_TSPA_9	
Assertion(s)	TA_TSPA_8	
Test Purpose	Verify that a trust list for the given trust scheme can be published successfully	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with *eidas.kamusm.gov.tr-example* should be published with *turkey.lightest.nlnetlabs.nl* domain name in TSPA	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl*	The service should return HTTP 200 OK with response data including eidas.kamusm.gov.tr-example scheme
2	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme turkey.lightest.nlnetlabs.nl PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_scheme._trust.turkey.lightest.nlnetlabs.nl. IN PTR ;; ANSWER SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl. 3600 IN PTR 1 1_scheme._trust.eidas.kamusm.gov.tr-example
3	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list with http://www.mindertestbed.org:8081/trust/TSL-XML.xml* parameter	The service should return HTTP 200 OK
4	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION:

		_scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www. mindertestbed.org:8081/trust/TSL-XML.xml*
5	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* SMIMEA ;; ANSWER SECTION: *_scheme._trust.eidas.kamusm.gov.tr-example... IN SMIMEA (3 0 1 0) with the full certificate*
6	Execute Minder-ATV *verifyTrustList* service that performs signature validation for the trust list downloaded in http://www. mindertestbed.org:8081/trust/TSL-XML.xml	Trust List validation result should be successfull
7	Execute Minder-ATV *checkCertificateFromSMIMEA* service that includes the certificate to be used during the validation of the trust list signer certificate	The certificate validation result should be successfull

ID	**TC_TSPA_10**	
Assertion(s)	TA_TSPA_8	
Test Purpose	Verify that trust list published on TSPA validation fails in case of invalid certificate constraints	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: with *eidas.kamusm.gov.tr-example* should be published with *turkey.lightest.nlnetlabs.nl* domain name in TSPA A valid trust list URI record is already defined on TSPA-DNS An SMIMEA record including an invalid certificate to be used in trust list validation	
Step	**Test Activity**	**Expected Result**

1	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www. mindertestbed.org:8081/trust/TSL-XML.xml*
2	On the terminal, type the following command: *dig _scheme._trust. /turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* SMIMEA ;; ANSWER SECTION: *_scheme._trust.eidas.kamusm.gov.tr-example... IN SMIMEA (3 0 1 0) with the full certificate*
3	Execute Minder-ATV *verifyTrustList* service that performs signature validation for the trust list downloaded in http://www. mindertestbed.org:8081/trust/TSL-XML.xml	Trust List validation result should be successfull
4	Execute Minder-ATV *checkCertificateFromSMIMEA* service that includes the certificate to be used during the validation of the trust list signer certificate	The certificate validation result should NOT be successfull

ID	**TC_TSPA_11**
Assertion(s)	TA_TSPA_9
Test Purpose	Verify that a Boolean trust scheme can be successfully defined and published on TSPA
Pre-Test Conditions	TSPA should already be deployed and running

	DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/ /turkey.lightest.nlnetlabs.nl/schemes with boolean.eidas.kamusm.gov.tr-example* parameter	The service should return HTTP 200 OK. (Total 2 trust scheme with *.eidas.kamusm.gov.tr-example and Boolean. .eidas.kamusm.gov.tr-example should be available)*
2	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *turkey.lightest.nlnetlabs.nl* PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ; QUESTION SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN* PTR ; ANSWER SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN PTR 1 1* *_scheme._trust.eidas.kamusm.gov.tr-example* *_scheme._trust.boolean.eidas.kamusm.gov.tr-example*
3	Minder sends to TSPA the following GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/ scheme/turkey.lightest.nlnetlabs.nl/schemes*	The service should return HTTP 200 OK where it contains the 3 scheme names: *eidas.kamusm.gov.tr-example* *Boolean. eidas.kamusm.gov.tr-example*

ID	**TC_TSPA_12**	
Assertion(s)	TA_TSPA_10	
Test Purpose	Verify that an ordinal trust scheme can be successfully defined and published on TSPA	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/ /turkey.lightest.nlnetlabs.nl/schemes with ordinallevelname.eidas.kamusm.gov.tr-example* parameter	The service should return HTTP 200 OK. (Total 3 trust scheme with *.eidas.kamusm.gov.tr-example and Boolean. .eidas.kamusm.gov.tr-example and ordinallevelname.eidas.kamusm.gov.tr-example should be available)*

2	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *turkey.lightest.nlnetlabs.nl* PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ; QUESTION SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN* PTR ; ANSWER SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN PTR 1 1* *_scheme._trust.eidas.kamusm.gov.tr-example* *_scheme._trust.boolean.eidas.kamusm.gov.tr-example* *_scheme._trust. ordinallevelname.eidas.kamusm.gov.tr-example*
3	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl/schemes* with no parameter	The service should return HTTP 200 OK where it contains the 3 scheme names: *eidas.kamusm.gov.tr-example* *Boolean. eidas.kamusm.gov.tr-example* *Ordinallevelname. eidas.kamusm.gov.tr-example*

ID	**TC_TSPA_13**	
Assertion(s)	TA_TSPA_10	
Test Purpose	Verify that trust schemes on TSPA can be successfully retrieved	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *turkey.lightest.nlnetlabs.nl* PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ; QUESTION SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN* PTR ; ANSWER SECTION:

		_scheme._trust.*turkey.lightest.nlnetlabs.nl IN PTR 1 1* *_scheme._trust.eidas.kamusm.gov.tr-example* *_scheme._trust.boolean.eidas.kamusm.gov.tr-example* *_scheme._trust. ordinallevelname.eidas.kamusm.gov.tr-example*
2	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl*	The service should return HTTP 200 OK where it contains the 3 scheme names: *eidas.kamusm.gov.tr-example* *Boolean. eidas.kamusm.gov.tr-example* *Ordinallevelname. eidas.kamusm.gov.tr-example*

ID	**TC_TSPA_14**	
Assertion(s)	TA_TSPA_10	
Test Purpose	Verify that received TSPA works in synchronization with DNS entries	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *Turkey.lightest.nlnetlabs.nl* PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ; QUESTION SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN* PTR ; ANSWER SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN PTR 1 1* *_scheme._trust.eidas.kamusm.gov.tr-example* *_scheme._trust.boolean.eidas.kamusm.gov.tr-example* *_scheme._trust. Ordinallevelname.eidas.kamusm.gov.tr-example*
2	On the terminal, login to DNS with ssh with ssh -i id_rsa tubitak@lightest.nlnetlabs.nl Goto /usr/home/zonemgr/etc folder Edit Open the zone file with the following command: "*vim lightest.nlnetlabs.nl*" Delete the corresponding record with turkey.lightest.nlnetlabs.nl and Ordinallevelname.eidas.kamusm.gov.tr-example trust scheme Close the ssh session	DNS should be updated.

3	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl*	The service should return HTTP 200 OK where it contains the 2 scheme names: *eidas.kamusm.gov.tr-example* *Boolean. eidas.kamusm.gov.tr-example*

ID	**TC_TSPA_15**	
Assertion(s)	TA_TSPA_11	
Test Purpose	Verify that delete service of TSPA for scheme name works successfuly	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *Turkey.lightest.nlnetlabs.nl* PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ; QUESTION SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN* PTR ; ANSWER SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN PTR 1 1* *_scheme._trust.eidas.kamusm.gov.tr-example* *_scheme._trust.boolean.eidas.kamusm.gov.tr-example*
2	Minder sends to TSPA the following HTTP DELETE request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/schemes*	The service should return HTTP 200 OK. *eidas.kamusm.gov.tr-example* *Boolean. eidas.kamusm.gov.tr-example* *Should be deleted.*
3	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *turkey.lightest.nlnetlabs.nl* PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1 ; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ; QUESTION SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN* PTR ; ANSWER SECTION:

ID	TC_TSPA_16	
Assertion(s)	TA_TSPA_11	
Test Purpose	Verify that delete service of TSPA for trust-list works successfully	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust scheme: *with eidas.kamusm.gov.tr-example* should be published with *turkey.lightest.nlnetlabs.nl* domain name in TSPA	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list with http://www.mindertestbed.org:8081/trust/TobedeletedTSL-XML.xml* trust list parameter	The service should return HTTP 200 OK
2	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www.mindertestbed.org:8081/trust/TobedeletedTSL-XML.xml*
3	Minder sends to TSPA the following HTTP DELETE request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list*	The service should return HTTP 200 OK
4	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION:

		; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION:

ID	**TC_TSPA_17**	
Assertion(s)	TA_TSPA_11	
Test Purpose	Verify that delete service of TSPA for trust-list works properly	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list with http://www.mindertestbed.org:8081/trust/TobedeletedTSL-XML.xml* trust list parameter	The service should return HTTP 200 OK
2	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www.mindertestbed.org:8081/trust/TobedeletedTSL-XML.xml*
3	Minder sends to TSPA the following HTTP DELETE request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list*	The service should return HTTP 200 OK
4	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

		;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION:

ID	**TC_TSPA_18**	
Assertion(s)	TA_TSPA_11	
Test Purpose	Verify that delete service of TSPA for a trust scheme works in synchronization with DNS records.	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/schemes with "tobedeleted.eidas.kamusm.gov.tr-example"* name	The service should return HTTP 200 OK
2	On the terminal, type the following command: *dig _scheme._trust.turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl *IN* PTR ;; ANSWER SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl IN PTR _scheme._trust. tobedeleted.eidas.kamusm.gov.tr-example
3	Minder sends to TSPA the following HTTP DELETE request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest /schemes*	The service should return HTTP 404 Not Found with reasonable error message should be displayed
4	On the terminal, type the following command: dig *_scheme._trust.turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer:

		;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl *IN* PTR ;; ANSWER SECTION: _scheme._trust.turkey.lightest.nlnetlabs.nl IN PTR

ID	**TC_TSPA_19**	
Assertion(s)	TA_TSPA_11	
Test Purpose	Verify that delete service of TSPA for trust-list works in synchronization with DNS records.	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list with http://www.mindertestbed.org:8081/trust/TobedeletedTSL-XML.xml* trust list parameter	The service should return HTTP 200 OK
2	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION: *;_scheme._trust.eidas.kamusm.gov.tr-exampl.e IN URI 1 1 http://www.mindertestbed.org:8081/trust/TobedeletedTSL-XML.xml*
3	Minder sends to TSPA the following HTTP DELETE request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list*	The service should return HTTP 200 OK

4	On the terminal, type the following command: *dig _scheme._trust. eidas.kamusm.gov.tr-example URI*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* URI ;; ANSWER SECTION:
5	Minder sends to TSPA the following HTTP DELETE request: https://lightest-dev.iaik.tugraz.at/tspa/api/v1/turkey.lightest.nlnetlabs.nl/trust-list	The service should return HTTP 404 Not Found with appropriate error message

ID	**TC_TSPA_20**	
Assertion(s)	TA_TSPA_12	
Test Purpose	Verify that tuple schemes can be defined and published on TSPA successfully.	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TSPA the following HTTP PUT request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/ /turkey.lightest.nlnetlabs.nl/schemes with tuplesnameandbirth.eidas.kamusm.gov.tr-example* parameter	The service should return HTTP 200 OK. (Total 1 trust scheme with *tuplesnameandbirth.eidas.kamusm.gov.tr-example should be available)*
2	On the terminal, type the following command: *dig _scheme._trust. turkey.lightest.nlnetlabs.nl PTR*	; <<>> DiG 9.10.6 <<>> _scheme._trust. *turkey.lightest.nlnetlabs.nl* PTR ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ; QUESTION SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN* PTR ; ANSWER SECTION: _scheme._trust.*turkey.lightest.nlnetlabs.nl IN PTR 1 1*

		scheme._trust.tuplesnameandbirth.eidas.kamusm.gov.tr-example
3	Minder sends to TSPA the following HTTP GET request: *https://lightest-dev.iaik.tugraz.at/tspa/api/v1/scheme/turkey.lightest.nlnetlabs.nl/schemes*	The service should return HTTP 200 OK where it contains the 1 scheme name: *tuplesnameandbirth.eidas.kamusm.gov.tr-example*

7.2 TTA

7.2.1 TTA Conformance Clauses

There is only one conformance clause:

CC_TTA_1: An implementation of TTA is conforming to TTA if it satisfies the conditions provided in the normative statements (NS_TTA_1-15) under the next section 7.2.2

7.2.2 TTA Normative Sources

Normative Sources given in D8.3 are not updated. For this reason, they will not be provided here.

7.2.3 TTA Test Assertions

TA ID	TA_TTA_1
Normative Source	NS_TTA_1, NS_TTA_9
Target	TTA-DNS
Prerequisite	The name and details (characteristics) of the trust scheme are defined in the TSPA and received from TSPA
Prescription Level	Mandatory
Predicate	Depending on the operating system that TTA is working on, the IP address of the DNS server exists and can be listed on the configurations and is already set on the TCP/IP Properties (DNS Server Address settings).

TA ID	TA_TTA_2
Normative Source	NS_TTA_1, NS_TTA_9
Target	TTA-DNS
Prerequisite	TA_TTA_1 The name and details (characteristics) of the trust scheme are defined in the TSPA and received from TSPA

Prescription Level	Mandatory
Predicate	TTA-DNS provides a means to secure DNS data by using digital signatures and public key cryptography. RR responses to the TTA translations query are signed by a valid Zone Key.

TA ID	TA_TTA_3
Normative Source	TA_NS_1, NS_TTA_2, TA_NS_3, NS_TTA_4, NS_TTA_10, NS_TTA_11
Target	TTA-Trust Translation Publisher
Prerequisite	The TTA DNS Name Server is up and running
Prescription Level	Mandatory
Predicate	TTA provides more than one translation schemes for the trust scheme.TTA provides/publishes a pointer to the trust translation list for each recognized trust level with Boolean, Ordinal or Tuple trust scheme types in the form of a series URI resource records.

TA ID	TA_TTA_4
Normative Source	NS_TTA_1, NS_TTA_3, NS_TTA_5, NS_TTA_7
Target	TTA-DNS
Prerequisite	The TTA DNS Name Server is up and running
Prescription Level	Mandatory
Predicate	For Boolean trust scheme, the prefixes for the TTA DNS record is set as "_translate" for the aspect and "_trust" for the application with the following format and the received DNS query form is:

```
;; QUESTION SECTION: Client/ATV to the TTA
;_translate._trust.etimestamp.eidas.eu.  IN  URI

;; ANSWER SECTION: from the TTA
_translate._trust.etimestamp.eidas.eu.   IN  URI
                    https://lightest.eu/ttl_qualifiedTimesta
                    mpEidas1.tpl
_translate._trust.etimestamp.eidas.eu.   IN  URI
                    https://lightest.eu/ttl_qualifiedTimesta
                    mpEidas1.xml
```

TA ID	TA_TTA_5
Normative Source	TA_NS_1, TA_NS_3, NS_TTA_4
Target	TTA-DNS
Prerequisite	The TTA DNS Name Server is up and running The names of the assurance levels just published by the TSPA have to be already retrieved from the TSPA by ATV

Prescription Level	Mandatory
Predicate	For Ordinal&Tuple Trust Scheme, the prefixes for the TTA DNS record are set as "_translate" for the aspect and "_trust" for the application and the assurance level (obtained from TSPA for the trust scheme) with the following format and the received DNS query form is :

```
;; QUESTION SECTION: Client/ATV to the TTA
;_translate._trust.qualified.eseal.eidas.eu.  IN  URI

;; ANSWER SECTION: from the TTA
_translate._trust.qualified.eseal.eidas.eu.   IN  URI
                    https://lightest.eu/ttl_qualifiedSealEid
                    as1.tpl

_translate._trust.qualified.eseal.eidas.eu.   IN  URI
                    https://lightest.eu/ttl_qualifiedSealEid
                    as1.xml
```

TA ID	TA_TTA_6
Normative Source	NS_ TTA_2, NS_ TTA_3, NS_ TTA_4, NS_ TTA_5
Target	TTA-Trust Translation Provider
Prerequisite	The TTA DNS Name Server is up and running
Prescription Level	Mandatory
Predicate	Trust Translation Provider provides a file for each recognized trust level with XML and TPL formats for the trust scheme. In case of XML, TTA returns the list of the trust levels equivalents to the one requested with level name and trust scheme name. In case of TPL, TTA returns the list of the trust levels equivalents to the one requested with level name, trust scheme name and TPL description.

TA ID	TA_TTA_7
Normative Source	NS_TTA_1, NS_TTA_3, NS_TTA_4, NS_TTA_8, NS_TTA_9
Target	TTA
Prerequisite	The TTA DNS Name Server is up and running
Prescription Level	Mandatory

Predicate	For Boolean trust scheme, TTA-DNS checks whether the certificate used for signing the translation files is valid according with the content of DNS-SMIMEA resource record. ;; QUESTION SECTION: Verifying authenticity ;_translate._trust.etimestamp.eidas.eu. IN **SMIMEA** ;; ANSWER SECTION: _translate._trust.etimestamp.eidas.eu. IN SMIMEA <SMIMEA record data>

TA ID	TA_TTA_8
Normative Source	NS_TTA_1, NS_TTA_3, NS_TTA_4, NS_ TTA_8, NS_TTA_9
Target	TTA
Prerequisite	The TTA DNS Name Server is up and running
Prescription Level	Mandatory
Predicate	For Ordinal&Tuple trust scheme, TTA-DNS checks whether the certificate used for signing the translation files is valid according with the content of DNS-SMIMEA resource record including the trust scheme and level of assurance ;; QUESTION SECTION: Verifying authenticity ;_translate._trust.**qualified**.eseal.eidas.eu. IN **SMIMEA** ;; ANSWER SECTION: _translate._trust.qualified.eseal.eidas.eu. IN SMIMEA <SMIMEA record data>

TA ID	TA_TTA_9
Normative Source	NS_ TTA_6, NS_ TTA_8, NS_TTA_9, NS_TTA_14
Target	TTA
Prerequisite	The TTA DNS Name Server is up and running Trust translation lists are already defined for the trust schemes The names of the assurance levels just published by the TSPA have to be already retrieved from the TSPA by ATV, in order to build the right domain name for asking for the translation.
Prescription Level	Mandatory
Predicate	Trust translation list documents, XML or TPL formats, are signed by the TTA with X.509 certificates.

TA ID	TA_TTA_10
Normative Source	NS_TTA_15
Target	TTA

Prerequisite	The TTA DNS Name Server is up and running The TTA should return the signed trust translation lists
Prescription Level	Mandatory
Predicate	TTA-DNS should provide certificate constraints to use for the verification of the translation list signature.

TA ID	TA_TTA_11
Normative Source	NS_TTA_10
Target	TTA
Prerequisite	The TTA DNS Name Server is up and running The TTA should return the signed trust translation lists
Prescription Level	Mandatory
Predicate	TTA should provide a delete service to delete translations

7.2.4 TTA Test Scenario

Turkey Trust Provider has negotiated with an EU General Trust Provider on whether their schemes trust each other, and in what way. The outcomes of these negotiations are then provided to the TTA, which represents the translation scheme for automated processing in LIGHT^{est}.This means the TTA becomes a function that allows the interoperability of trust schemes published by different entities, even across different trust domains, by defining the relation between the trust scheme levels.

7.2.4.1 Discovery of the translation list for a boolean trust scheme

Agreement details:

Source Trust Scheme: *timestamp.eidas.kamusm.gov.tr-example* with qualified assurance level Target Trust Scheme: *timestamp.eidas.eu* with qualified assurance level. For this translation, there exists a translation list with TPL and XML formats on TTA. The translation is defined as follows:

{ "agreement": {

"name": "test-agreement","status": "active",

"creation-date": "2018-08-10",

"leaving-date": "2019-08-10",

```
"activation-date": "2018-08-10",
"source": {
        "level": "qualified",
        "name": " timestamp.eidas.kamusm.gov.tr-example",
        "provider": "Turkey Trust Provider",
        "params": [{
                "name": "param1-name",
                "value": "param1-value" },
            { "name": "param2-name",
              "value": "param2-value"} ]},
"target": {
        "level": "qualified",
        "name": " timestamp.eidas.eu ",
        "provider": "EU Trust Provider",
        "params": [{
                "name": "param3-name",
                "value": "param3-value"},
            {  "name": "param4-name",
            "value": "param5-value"}]}} }
```


A translation with an invalid translation list is defined on TTA “invalid-agreement” with the following translation lists: *http://www.mindertestbed.org:8081/ttl/ttl_invalidqualifiedTimestampEidas1.tpl* and *http://www.mindertestbed.org:8081/ttl/ttl_invalidqualifiedTimestampEidas1.xml*

7.2.4.2 Discovery of the translation list for an ordinal trust scheme

The electronic eSeal trust service in the eIDAS trust scheme provides two levels of trust, namely, Advanced and Qualified.

A translation called “test-agreement-ordinal” translation is defined on TTA. The URI of the translation is *http://www.mindertestbed.org:8081/tta/*eIDAS_eSeal.xml

A query is sent to the TTA to know about its equivalent levels of **eseal*.eidas.kamusm.gov.tr-example*** for qualified level name in other trust schemes. The verifier needs to check this claim by locating the trust translation declaration.

The Evidence Record trust service in the eIDAS trust scheme provides two levels of trust, namely, High and Low.

A query is sent to the TTA to know about its equivalent levels of **evidence*.eidas.kamusm.gov.tr-example*** for qualified level name in other trust schemes. The verifier needs to check this claim by locating the trust translation declaration.

Since the contents of the files are signed, SMIMEA query is sent for certificate validation check.

A translation with an invalid translation list is defined on TTA “invalid-agreement-ordinal” with the following translation lists: http://www.mindertestbed.org:8081/ttl/ttl_invalidlowevidenceEidas1.tpl and http://www.mindertestbed.org:8081/ttl/ttl_invalidlowevidenceEidas1.xml

7.2.4.3 Discovery of the translation list for a tuple trust scheme

Definition of tuple-based trust scheme in the TSPA should be: (attribute name, attribute value)
A new “test-agreement-tuple” translation is defined.

STORK AQAA assigns an attribute quality assurance level to a group of attributes provides as part of an electronic ID. These assurance levels are derived from both the quality assurance level of the eID itself as well as the maximum of the quality assurance levels of each of the attributes in the group. (1-4 AQAA Level)

A fictional eID scheme eid.kamusm.gov.tr-example defines an attribute group name-and-year-of-birth that contains the attributes for the name and year of birth of the holder of the eID.

It constructs the domain name to query as _translate._trust.name-and-year-of-birth.kamusm.gov.tr-example and queries for URI records.

A translation with an invalid translation list is defined on TTA “invalid-agreement-tuple” with the following translation lists: *http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-*

1.tpl and *http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.xml*

7.2.5 TTA Test Cases

This section includes the test case list and test case details.

TTA Test cases assume that scheme information is obtained from TSPA and electronic transaction is parsed on Minder-ATV and is sent to TSPA to conform the trust scheme membership.

7.2.5.1 TTA Test Case List

Table 2 List of TTA test cases

ID	Purpose
TC_TTA_1	Check if TSPA-DNS server exists and is configured to be used by TTA.
TC_TTA_2	Verify that RR responses are signed by a valid Zone Key of DNS
TC_TTA_3	Verify that TTA published more than one translation schemes for a boolean trust scheme
TC_TTA_4	Verify that TTA published more than one translation schemes for an ordinal trust scheme
TC_TTA_5	Verify that TTA published more than one translation schemes for a tuple trust scheme
TC_TTA_6	Verify that RR response to the issuername is a URI record for a boolean scheme and DNSSec validation is successfull
TC_TTA_7	Verify that RR response to the issuername is a URI record for ordinal schemes and DNSSec validation is successfull
TC_TTA_8	Verify that TTA published more than one translation schemes for a tuple trust scheme
TC_TTA_9	Check that the verification result of translation list signature is successfull for a boolean trust scheme
TC_TTA_10	Check that the verification result of translation list signature is successfull for a ordinal trust scheme
TC_TTA_11	Check that the verification result of translation list signature is successfull for a tuple trust scheme
TC_TTA_12	Verify that an invalid trust list,pointed on the URI RR record, signature validation fails for boolean trust scheme
TC_TTA_13	Verify that an invalid trust list,pointed on the URI RR record, signature validation fails for ordinal trust scheme
TC_TTA_14	Verify that an invalid trust list,pointed on the URI RR record, signature validation fails for tuple trust scheme
TC_TTA_15	Verify that the certificate provided by DNS is not valid and translation list verification fails due to certificate validation
TC_TTA_16	Verify that received TTA works in synchronization with DNS entries
TC_TTA_17	Verify that delete service of TTA for scheme name works properly
TC_TTA_18	Verify that delete service of TTA for scheme name works properly
TC_TTA_19	Verify that delete service of TTA for scheme name works properly

7.2.5.2 TTA Test Case Details

ID	TC_TTA_1
Assertion(s)	TA_TTA_1
Test Purpose	Check if TSPA-DNS server exists and is configured to be used by TTA.
Pre-Test Conditions	TTA should already be deployed DNS deployment should be available Test Inputs are generated accordng to Test Scenario for TTA section

Step	Test Activity	Expected Result
1	On the terminal, type the following command: *dig lightest.nlnetlabs.nl*	; <<>> DiG 9.10.6 <<>> lightest.nlnetlabs.nl ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8688 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;lightest.nlnetlabs.nl. IN A ;; ANSWER SECTION: lightest.nlnetlabs.nl. 3600 IN A 185.49.141.61
2	Minder sends to TTA the following HTTP PUT request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/createTranslation* *with "test-agreement" Boolean trust scheme*. Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK
4	On the terminal, type the following command: *dig _translate._trust.turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust.turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl*

		http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*

ID	**TC_TTA_2**	
Assertion(s)	TA_TTA_2	
Test Purpose	Verify that RR responses are signed by a valid Zone Key of DNS	
Pre-Test Conditions	TTA DNS is up and running with DANE protocol enabled in the configured DNS	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig lightest.nlnetlabs.nl +noall +comments*	; <<>> DiG 9.10.6 <<>> lightest.nlnetlabs.nl +noall +comments ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10833 ;; flags: qr rd ra aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

ID	**TC_TTA_3**	
Assertion(s)	TA_TTA_3	
Test Purpose	Verify that TTA published more than one translation schemes for a boolean trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TTA the following HTTP PUT request: http://tta-*lightest.eu:8080/ttaFM/mng/rsc/createTranslation* *with "test-agreement"* Boolean trust scheme. Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK
2	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* *with "test-agreement"*	The service should return HTTP 200 OK. The return JSON value should include the translation information given in the Test Scenarion for TTA "test-agreement"
3	On the terminal, type the following command: *dig _translate._trust.turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust.turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer:

		;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*

ID	**TC_TTA_4**	
Assertion(s)	TA_TTA_3	
Test Purpose	Verify that TTA published more than one translation schemes for an ordinal trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TTA the following HTTP PUT request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/createTranslation* with “test-agreement-ordinal” Ordinal trust scheme. Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK
2	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* *with “test-agreement-ordinal”*	The service should return HTTP 200 OK. The return JSON value should include the translation information given in the Test Scenarion for TTA “test-agreement-ordinal”
3	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761

		;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.tpl …* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.tpl* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.xml* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.xml*

ID	**TC_TTA_5**	
Assertion(s)	TA_TTA_3	
Test Purpose	Verify that TTA published more than one translation schemes for a tuple trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TTA the following HTTP PUT request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/createTranslation with "test-agreement-tuple" tuple trust scheme.* Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK
2	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation with "test-agreement-ordinal"*	The service should return HTTP 200 OK. The return JSON value should include the translation information given in the Test Scenario for TTA "test-agreement-tuple"
3	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer:

		;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.xml* *_translate._trust.name-and-year-of-birth.kamusm.gov.tr-example. IN URI* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.tpl* *_translate._trust.name-and-year-of-birth.kamusm.gov.tr-example. IN URI* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.xml* *_translate._trust.name-and-year-of-birth.kamusm.gov.tr-example. IN URI* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.tpl*

ID	**TC_TTA_6**	
Assertion(s)	TA_TTA_4	
Test Purpose	Verify that RR response to the issuername is a URI record for a boolean scheme and DNSSec validation is successfull	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running Published trust translation list declarations are available for “test-agreement“	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* with with “test-agreementagreement” boolean trust scheme. Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK and JSON return value is the agreement details given in Test Scenario for TTA. The result should include the following translation lists: *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml*

		http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml
2	On the terminal, type the following command: *dig _translate._trust.turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust.turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*

ID	**TC_TTA_7**	
Assertion(s)	TA_TTA_5, TA_TTA_6	
Test Purpose	Verify that RR response to the issuername is a URI record for ordinal schemes and DNSSec validation is successfull	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running Published trust translation list declarations are available for “test-agreement-ordinal“	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* with “test-agreement-ordinal” Ordinal trust scheme. Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK. The result should include the following translation lists: *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.xml*

		http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.xml
2	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation with "test-agreement-ordinal"*	The service should return HTTP 200 OK. The return JSON value should include the translation information given in the Test Scenarion for TTA "test-agreement-ordinal"
3	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.tpl …* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr- example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.tpl* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr- example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.xml* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr- example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.xml*

ID	**TC_TTA_8**	
Assertion(s)	TA_TTA_5, TA_TTA_6	
Test Purpose	Verify that TTA published more than one translation schemes for a tuple trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running Published trust translation list declarations are available for "test-agreement-tuple"	
Step	**Test Activity**	**Expected Result**

1	Minder sends to TTA the following *HTTP GET request:* *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* with “test-agreement-tuple” tuple trust scheme. Translation definition is given in Test Scenario for TTA	The service should return HTTP 200 OK. The following translation files should be included: *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.xml* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.tpl* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.xml* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.tpl*
2	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* *with “test-agreement-tuple”*	The service should return HTTP 200 OK. The return JSON value should include the translation information given in the Test Scenarion for TTA “test-agreement-tuple”
3	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.xml* *_translate._trust.name-and-year-of-birth.kamusm.gov.tr-example. IN URI* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.tpl* *_translate._trust.name-and-year-of-birth.kamusm.gov.tr-example. IN URI* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.xml* *_translate._trust.name-and-year-of-birth.kamusm.gov.tr-example. IN URI* *http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.tpl*

ID	TC_TTA_9	
Assertion(s)	TA_TTA_10, TTA_9	
Test Purpose	Check that the verification result of translation list signature is successfull for a boolean trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running A valid translation with test-agreement scheme name is defined on TTA	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*
2	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* for the following translation lists http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml	The trust lists should be downloaded and should be opened via an XML editor
3	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return TRUE

4	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _translate._trust. turkey.lightest.nlnetlabs.nl *IN* SMIMEA ;; ANSWER SECTION: *_translate_trust. turkey.lightest.nlnetlabs.nl IN SMIMEA (3 0 1 0) with the full certificate*

ID	**TC_TTA_10**	
Assertion(s)	TA_TTA_10, TA_TTA_7, TA_TTA_9	
Test Purpose	Check that the verification result of translation list signature is successfull for a ordinal trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running A valid translation with test-agreement-ordinal scheme name is defined on TTA	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.tpl …* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.tpl* *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr-example IN URI*

		http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.xml *_translate._trust.qualified.eseal.eidas.kamusm.gov.tr- example IN URI* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.xml*
2	Minder-ATV parses the DNS query and Execute Minder-ATV downloadservice for the following translation lists: http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidas1.xml http://www.mindertestbed.org:8081/ttl/ttl_qualifiedSealEidasN.xml	The trust lists should be downloaded and should be opened via an XML editor
3	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return TRUE
4	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _translate._trust.turkey.lightest.nlnetlabs.nl IN SMIMEA ;; ANSWER SECTION: *_translate_trust. turkey.lightest.nlnetlabs.nl IN SMIMEA (3 0 1 0) with the full certificate*

ID	**TC_TTA_11**	
Assertion(s)	TA_TTA_10, TA_TTA_8	
Test Purpose	Check that the verification result of translation list signature is successfull for a tuple trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running A valid translation with test-agreement-tuple scheme name is defined on TTA	
Step	**Test Activity**	**Expected Result**

1	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI
2	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* for the following translation lists : http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.xml http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-1.tpl http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.xml http://www.mindertestbed.org:8081/ttl/name-and-year-of-birth/ttl-2.tpl	The trust lists should be downloaded and should be opened via an XML editor
3	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return TRUE
4	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _translate._trust.turkey.lightest.nlnetlabs.nl IN SMIMEA ;; ANSWER SECTION: *_translate_trust. turkey.lightest.nlnetlabs.nl IN SMIMEA (3 0 1 0) with the full certificate*

ID	TC_TTA_12
Assertion(s)	TA_TTA_9
Test Purpose	Verify that an invalid trust list,pointed on the URI RR record, signature validation fails for boolean trust scheme
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running There exists an translation agreement “invalid-agreement“ that includes an invalid trust list for a boolean trust scheme

Step	Test Activity	Expected Result
1	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* *with “invalid-agreement” parameter*	The service should return HTTP 200 OK. The following translations should be listed: *http://www.mindertestbed.org:8081/ttl/ttl_invalidqualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_invalidqualifiedTimestampEidas1.xml*
2	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *invalidqualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_invalidqualifiedTimestampEidas1.xml*
3	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* with the following files: *invalidqualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_invalidqualifiedTimestampEidas1.xml*	The trust list should be downloaded and should be opened via an XML editor
4	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return FALSE

5	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> *DiG 9.10.6 _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA* *;; global options: +cmd* ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _translate._trust.turkey.lightest.nlnetlabs.nl *IN* SMIMEA ;; ANSWER SECTION: *_translate._trust turkey.lightest.nlnetlabs.nl IN SMIMEA (3 0 1 0) with the full certificate*
6	Execute Minder-ATV *checkCertificatefromSMIMEA* service to verify the certificate used to sign the trust list	Certificate validation result should NOT be successfull

ID	**TC_TTA_13**	
Assertion(s)	TA_TTA_9, TA_TTA_07	
Test Purpose	Verify that an invalid trust list,pointed on the URI RR record, signature validation fails for ordinal trust scheme	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running There exists an translation agreement “invalid-agreement-ordinal“ that includes an invalid trust list for a ordinal trust scheme	
Step	**Test Activity**	**Expected Result**
1	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* *with “invalid-agreement-ordinal” parameter*	The service should return HTTP 200 OK. The following translations should be listed: *http://www.mindertestbed.org:8081/ttl/ttl_invalidloweviden ceEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_invalidloweviden ceEidas1.xml*
2	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI

		;; ANSWER SECTION: *http://www.mindertestbed.org:8081/ttl/ttl_invalidlowevidenceEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_invalidlowevidenceEidas1.xml*
3	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice* with the following files: *http://www.mindertestbed.org:8081/ttl/ttl_invalidlowevidenceEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_invalidlowevidenceEidas1.xml*	The trust list should be downloaded and should be opened via an XML editor
4	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return FALSE
5	On the terminal, type the following command: *dig _translate._trust.turkey.lightest.nlnetlabs.nl SMIMEA*	; *<<>> DiG 9.10.6 _translate._trust.turkey.lightest.nlnetlabs.nl SMIMEA* *;; global options: +cmd* ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _translate._trust.turkey.lightest.nlnetlabs.nl *IN* SMIMEA ;; ANSWER SECTION: *_translate._trust turkey.lightest.nlnetlabs.nl IN SMIMEA (3 0 1 0) with the full certificate*
6	Execute Minder-ATV *checkCertificatefromSMIMEA* service to verify the certificate used to sign the trust list	Certificate validation result should NOT be successful,

ID	**TC_TTA_14**		
Assertion(s)	TA_TTA_9, TA_TTA_08		
Test Purpose	Verify that an invalid trust list,pointed on the URI RR record, signature validation fails for tuple trust scheme		
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running There exists an translation agreement “invalid-agreement-tuple“ that includes an invalid trust list for a tuple trust scheme		
Step	**Test Activity**		**Expected Result**

1	Minder sends to TTA the following *HTTP GET request: http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation with "invalid-agreement-tuple" parameter*	The service should return HTTP 200 OK. The following translations should be listed: *http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.tpl* *http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.xml*
2	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: *http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.tpl* *http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.xml*
3	Minder-ATV parses the DNS query and Execute Minder-ATV *downloadservice with the following files:* *http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.tpl* *http://www.mindertestbed.org:8081/ttl/invalidname-and-year-of-birth/ttl-1.xml*	The trust list should be downloaded and should be opened via an XML editor
4	Execute Minder-ATV *verifyTrustList* service that performs signature validation	The trust list verification should return FALSE
5	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA*	; <<>> *DiG 9.10.6 _translate._trust. turkey.lightest.nlnetlabs.nl SMIMEA* *;; global options: +cmd* ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION:

		; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _translate._trust.turkey.lightest.nlnetlabs.nl *IN* SMIMEA ;; ANSWER SECTION: *_translate._trust turkey.lightest.nlnetlabs.nl IN SMIMEA (3 0 1 0) with the full certificate*
6	Execute Minder-ATV *checkCertificatefromSMIMEA* service to verify the certificate used to sign the trust list	Certificate validation result should NOT be successful,

ID	**TC_TTA_15**	
Assertion(s)	TA_TTA_09, TA_TTA_08	
Test Purpose	Verify that the certificate provided by DNS is not valid and translation list verification fails due to certificate validation	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A valid trust list URI record is already defined on TSPA-DNS An SMIMEA record including an invalid certificate to be used in translation list validation exists on the DNS	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*
2	On the terminal, type the following command:	; <<>> DiG 9.10.6 <<>> _scheme._trust. *eidas.kamusm.gov.tr-example* SMIMEA ;; global options: +cmd

	dig _scheme._trust. /turkey.lightest.nlnetlabs.nl SMIMEA	;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: _scheme._trust. *eidas.kamusm.gov.tr-example IN* SMIMEA ;; ANSWER SECTION: *_scheme._trust.eidas.kamusm.gov.tr-example... IN SMIMEA (3 0 1 0) with the full certificate*
3	Execute Minder-ATV *verifyTrustList* service that performs signature validation for the trust list downloaded in http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml	Trust List validation result should be successfull
4	Execute Minder-ATV *checkCertificateFromSMIMEA* service that includes the certificate to be used during the validation of the trust list signer certificate	The certificate validation result should NOT be successfull

ID	**TC_TTA_16**	
Assertion(s)	TA_TTA_1, TA_TTA_2	
Test Purpose	Verify that received TTA works in synchronization with DNS entries	
Pre-Test Conditions	TTA should already be deployed and running DNS deployment with DNSSec extension should be running	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI for "test-agreement"*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

		;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*
2	On the terminal, login to DNS with ssh with ssh -i id_rsa tubitak@lightest.nlnetlabs.nl Goto /usr/home/zonemgr/etc folder Edit Open the zone file with the following command: *vim lightest.nlnetlabs.nl* Delete the corresponding record with turkey.lightest.nlnetlabs.nl and http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.tpl http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.tpl Close the ssh session	DNS should be updated.
3	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation*	The service should return HTTP 200 OK where it contains the 2 translation files *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*

ID	**TC_TTA_17**	
Assertion(s)	TA_TSPA_7	
Test Purpose	Verify that delete service of TTA for scheme name works properly	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A translation with “test-agreement“ should be already defined on TTA	
Step	**Test Activity**	**Expected Result**

1	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI for "test-agreement"*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*
2	Minder sends to TTA the following HTTP DELETE request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/deleteTranslation* *for "test-agreement" translation*	The service should return HTTP 200 OK.
3	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI for "test-agreement"*	; <<>> DiG 9.10.6 <<>> dig _translate._trust. turkey.lightest.nlnetlabs.nl URI ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1 ; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ; QUESTION SECTION: dig _translate._trust. turkey.lightest.nlnetlabs.nl IN URI ; ANSWER SECTION:

ID	**TC_TTA_18**	
Assertion(s)	TA_TTA_11	
Test Purpose	Verify that delete service of TTA for scheme name works properly	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A translation with "test-agreement" should be already defined on TTA	
Step	**Test Activity**	**Expected Result**

1	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI for "test-agreement"*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*
2	Minder sends to TTA the following HTTP DELETE request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/deleteTranslation* *for "test-agreement" translation*	The service should return HTTP 200 OK.
3	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI for "test-agreement"*	turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: ; ANSWER SECTION:
4	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation* *for for "test-agreement" translation*	The service should return HTTP 404 Not Found.

ID	TC_TTA_19	
Assertion(s)	TA_TTA_11	
Test Purpose	Verify that delete service of TTA for scheme name works properly	
Pre-Test Conditions	TSPA should already be deployed and running DNS deployment with DNSSec extension should be running A translation with “test-agreement“ should be already defined on TTA	
Step	**Test Activity**	**Expected Result**
1	On the terminal, type the following command: *dig _translate._trust. turkey.lightest.nlnetlabs.nl URI for “test-agreement”*	; <<>> DiG 9.10.6 <<>> _translate._trust. turkey.lightest.nlnetlabs.nl URI; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53761 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags:; udp: 4096 ;; QUESTION SECTION: ;_translate._trust.turkey.lightest.nlnetlabs.nl. IN URI ;; ANSWER SECTION: _ translate._trust.turkey.lightest.nlnetlabs.nl. 3600 IN URI *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidas1.xml* *http://www.mindertestbed.org:8081/ttl/ttl_qualifiedTimestampEidasN.xml*
2	Minder sends to TTA the following HTTP DELETE request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/deleteTranslation for “test-agreement-dummy” translation*	The service should return HTTP 404 Not Found.
3	Minder sends to TTA the following HTTP GET request: *http://tta-lightest.eu:8080/ttaFM/mng/rsc/getTranslation for for “test-agreement-dummy” translation*	The service should return HTTP 404 Not Found.

7.3 DP

In order to test if the Delegation Provider (DP) implementation conforms to the DP specifications, conformance clauses, the normative statements and the test assertions are given below.

7.3.1 DP Conformance Clauses

CC_DP_1: An implementation of DP is conforming to DP if it satisfies the conditions provided in the normative statements NS_DP_1 to NS_DP_13.

7.3.2 DP Normative Statements

Normative Sources for DP given in D8.3 are not updated. For this reason, they will not be provided here.

7.3.3 DP Test Assertions

Test Assertions for DP given in D8.3 are not updated. For this reason, they will not be provided here.

7.3.4 DP Test Scenario

TUBITAK wants to empower an employee (TUBITAK Tester) to do purchasing tasks on behalf of the company. The employee shall only have the allowance to do purchases up to a certain amount. All purchases above that amount require the authorization of the manager of the company. The employee receives the delegation for a special purpose, as the employee will be able to do purchases on behalf of the company.

First of all, the Mandator who is the manager of the company needs to create the delegation on Delegation Provider GUI. The delegation contains the following:

1. delegation type,
2. proxy (employee) public key,
3. mandator private key and mandator public key,
4. validity time,
5. address of delegation provider and the attributes the Mandator wants to transfer to the Proxy.

7.3.4.1 Test Scenario for DP (publication & download services):

The Mandator needs to sign the delegation file with his private key. Signed delegation file is encrypted by mandators' symmetric key. Also, symmetric key is encrypted by proxy's public key. Encrypted delegation and encrypted symmetric key are published on the Delegation Provider at https://mindertestbed.org/keys. Publication service returns http status to Mandator. This bilateral delegation is between a Mandator and a Proxy. The structure of a delegation in XML :

```
<?xml version="1.0"?>
<delegation version="1.0">
        <!-- Mandatory Information -->
        <issuedDate> 2017-05-14T23:59:59 </issuedDate>
        <proxy> TUBITAK Tester </proxy>
        <issuer> Tubitak </issuer>
        <bilateral />
        <substitutionAllowed>false</substitutionAllowed>
        <delegationAllowed>false</delegationAllowed>
        <validity>
                <notBefore> 2019-05-15T00:00:00 </notBefore>
                <notAfter> 2020-15-15T23:59:59 </notAfter>
        </validity>
        <domain name="purchase" version="0">
        </domain>
        <ds:signature>
```

```
        </ds:signature>
</delegation>
```

In the second step, the employee receives the delegation for a special purpose, as the employee will be able to do purchases on behalf of the company. For the client to find the delegation, the delegation information has to be included in the transaction itself. The Proxy provides the delegation information for the transaction. Also, proxy needs the symmetric key to decrypt the delegation. To do so, the Proxy needs to download the encrypted delegation and encrypted symmetric key from the Delegation Provider.

Proxy knows (assumption) that Delegation Provider is located at https://mindertestbed.org/ and called as dpUrlAddress throughout the test cases.

TUBITAK Tester delegation is published in
https://mindertestbed.org/delegation/tubitak_delegation.xml

TUBITAK Tester invalid delegation is published in
https://mindertestbed.org/delegation/invalidtubitak_delegation.xml

Proxy sends following http request to DP-downloadService to download the delegation:
https://mindertestbed.org/download/{id}?token=xxxxxxxxxx

DP-downloadService returns a response which contains encrypted delegation in XML format to proxy.

Proxy sends following HTTP request to DP-downloadService to download the symmetric key:
https://mindertestbed.org/download_key/{id}?token=xxxx

DP-downloadService returns a response which contains encrypted symmetric key to proxy.

Proxy decrypts symmetric key with proxy's public key and then decrypts delegation file with symmetric key.

Third step, eTransaction which contains an invoice and delegation information embedded in ASIC-S container prepared. Verifier (company) sends eTransaction to Client. Client gets the URL which is the address of Delegation Provider from delegation.xml

Then, Client sends following http request to DP- searchService to find out status of delegation:
mindertestbed.org/download/{id}?token=xxxxxxxxxx

DP-searchService returns a response that the delegation is not revoked.

7.3.4.2 Test Scenario for DP (revocation & search services)

An employee leaves the company. The employee holds a delegation for purchasing purposes. Now the company has to revoke the delegation because the employee does not have an allowance to do purchases on behalf of company anymore.

First step for revoke the delegation, Mandator search the delegation on DP to ensure that the delegation is correct.

Mandator sends following HTTP request to DP- searchService to find out status of delegation:
https://mindertestbed.org/download/{id}?token=xxxxxxxxxx

7.3.5 DP Test Cases

This section includes the test case list and test case details

7.3.5.1 DP Test Case List

Table 3 List of DP test cases

ID	Purpose
TC_DP_1	Check the response when DP publishes the delegation.
TC_DP_2	Verify that DP publishes a delegation successfully.
TC_DP_3	Verify that DP publishes a delegation successfully.
TC_DP_4	Check the response when DP verifies the delegation.
TC_DP_5	Check content of delegation from the response when delegation is verified.
TC_DP_6	Check the response when revoked delegation is queried
TC_DP_7	Check the response when valid delegation is queried
TC_DP_8	Check the response that DP should return error if verifier sends more than one revocation query at the time
TC_DP_9	Check if a revoke command interface is available
TC_DP_10	Check revocation response when verifier sends a revocation query. Check the signed revocation response with the certificate that is issued by Mandator for the revocation purpose. Check if delegation id is hash of delegation. Check if the response includes the delegation that is given to DP, the certificates that is used to sign and all certificates to build the trust chain.
TC_DP_11	Check if delegation id is hash of delegation.
TC_DP_12	Publish delegation key
TC_DP_13	Download delegation key

7.3.5.2 DP Test Case Details

ID	TC_DP_1
Assertion(s)	TA_DP_1, TA_DP_2
Test Purpose	Check the response when DP publishes the delegation.
Pre-Test Conditions	Delegation Provider is accessible.

	Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester dpUrlAddress: to be defined when DP deployment is completed	
Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends to DP the following HTTP POST request: *https://**dpUrlAddress**/1/publish with delegation.xml (delegation.xml, pk and publicKeyHash) parameters*	The service should return HTTP 201. The response data should include the receipt including delegation type, Mandator's private key, Proxy public key, validity time, domain settings and address of DP.
2	Execute Minder-ATV *downloadservice* with the following parameter: delegation URL	https://mindertestbed.org/delegation/tubitak_delegation.xml should be downloded and opened in an XML editor

ID	**TC_DP_2**	
Assertion(s)	TA_DP_1, TA_DP_2	
Test Purpose	Verify that DP publishes a delegation successfully.	
Pre-Test Conditions	Delegation Provider (DP) is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester dpUrlAddress: to be defined when DP deployment is completed	
Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends to DP the following HTTP POST request: *https://**dpUrlAddress**/1/publish with delegation.xml (invaliddelegation.xml, pk and publicKeyHash) parameters*	The service should return HTTP 500.

ID	**TC_DP_3**
Assertion(s)	TA_DP_1
Test Purpose	Verify that DP publishes a delegation successfully.
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester dpUrlAddress: to be defined when DP deployment is completed

	It is assumed that DP does not publish delegation in this case.

Step	Test Activity	Expected Result
1	Minder-ATV sends to DP the following HTTP POST request: *https://**dpUrlAddress**/1/publish with delegation.xml (delegation.xml, pk and publicKeyHash) parameters* Note that publicKeyHash does not belong to Proxy	The service should return HTTP 500

ID	TC_DP_4	
Assertion(s)	TA_DP_1, TA_DP_2, TA_DP_3, TA_DP_6	
Test Purpose	Check the response when DP verifies the delegation.	
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP dpUrlAddress: to be defined when DP deployment is completed	
Step	**Test Activity**	**Expected Result**
1	Mandator creates a delegation for TUBITAK Tester.	Delegation.xml that conforms to ETSI 119 621 is created
2	Execute Minder-ATV *verifydelegationservice* with the following parameter: delegation.xml	The expected result is True
2	Minder-ATV sends to DP the following HTTP POST request: *https://**dpUrlAddress**/1/publish with delegation.xml (delegation.xml, pk and publicKeyHash) parameters*	The service should return HTTP 201. The response data should include the receipt including delegation type, Mandator's private key, Proxy public key, validity time, domain settings and address of DP.

ID	TC_DP_5
Assertion(s)	TA_DP_7
Test Purpose	Check content of delegation from the response when delegation is verified.

Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester dpUrlAddress: to be defined when DP deployment is completed	
Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends to DP the following HTTP POST request: *https://**dpUrlAddress**/1/publish with delegation.xml (delegation.xml, pk and publicKeyHash) parameters*	The service should return HTTP 201. The response data should include the receipt including validity time, Sequence Number, IssuedDate, Proxy, Mandator, validity, notAfter, notBefore, flags, server fields.
2	Execute Minder-ATV *downloadservice* with the following parameter: delegation URL	https://mindertestbed.org/delegation/tubitak_delegation.xml should be downloded and opened in an XML editor

ID	**TC_DP_6**	
Assertion(s)	TA_DP_8	
Test Purpose	Check the response when revoked delegation is queried	
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester dpUrlAddress: to be defined when DP deployment is completed A Revoked delegation for TUBITAK Tester is published in DP	
Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends HTTP request below to DP searchServer to query status of delegation *https://dpUrlAddress/search?delegation{id}?&token={token}*	DP service sends a response that the delegation is revoked. To be defined when implemented.
2	Minder-ATV executes *verifyRevocationResponse service to validate the revocation result*	The verification result should be successful.

ID	**TC_DP_7**

Assertion(s)	TA_DP_9	
Test Purpose	Check the response when valid delegation is queried	
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester dpUrlAddress: to be defined when DP deployment is completed A valid delegation for TUBITAK Tester is published in DP	
Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends HTTP request below to DP searchServer to query status of delegation https://dpUrlAddress/search?delegation{id}?&token={token}	DP service sends a response that the delegation is valid. To be defined when implemented.

ID	**TC_DP_8**	
Assertion(s)	TA_DP_10	
Test Purpose	Check the response that DP should return error if verifier sends more than one revocation query at the time	
Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester dpUrlAddress: to be defined when DP deployment is completed A valid delegation for TUBITAK Tester is published in DP	
Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends two revocation query at the same time. https://dpUrlAddress/search?delegation{idbfgbfgbgf}?&token={token} https://dpUrlAddress/search?delegation{iddddd}?&={token}	Service returns error message. To be defined when implemented.

ID	**TC_DP_9**
Assertion(s)	TA_DP_11
Test Purpose	Check if a revoke command interface is available

Pre-Test Conditions	Delegation Provider is accessible. Delegation is prepared by TUBITAK as Mandator as a signed and encrypted delegation in XML format. Details of the delegation is defined in Test Scenario for DP Proxy is TUBITAK Tester dpUrlAddress: to be defined when DP deployment is completed A valid delegation for TUBITAK Tester is published in DP	
Step	**Test Activity**	**Expected Result**
1	Mandator sends HTTP request below to DP searchServer to query status of delegation *https://dpUrlAddress/search?delegation{id}?&token={id}*	DP service sends a response that the delegation is valid. To be defined when implemented.
2	Mandator sends delegation Id (hash of delegation), certificate signed by mandator and revocation delegation to DP	DP revokes the delegation and stores the revocation time with the revocation. To be defined when implemented.
3	Minder-ATV executes *verifyRevocationResponse* service to validate the revocation response	Verification of revocation should be successfull

ID	**TC_DP_10**	
Assertion(s)	TA_DP_12, TA_DP_13, TA_DP_14, TA_DP_15	
Test Purpose	Check revocation response when verifier sends a revocation query. Check the signed revocation response with the certificate that is issued by Mandator for the revocation purpose. Check if delegation id is hash of delegation. Check if the response includes the delegation that is given to DP, the certificates that is used to sign and all certificates to build the trust chain.	
Pre-Test Conditions	Delegation Provider is accessible. Revoked delegation file is available.	
Step	**Test Activity**	**Expected Result**
1	Verifier sends HTTP request below to DP searchServer to query status of delegation *https://dpUrlAddress/search?delegation{id}?&token={id}*	DP searchs revocation archieve and prepares the response. To be defined when implemented.

2	DP sends response which is signed.	Verifier checks the response. To be defined when implemented.

ID	**TC_DP_11**	
Assertion(s)	TA_DP_14	
Test Purpose	Check if delegation id is hash of delegation.	
Pre-Test Conditions	Delegation Provider is accessible. Delegation is not available on DP	
Step	**Test Activity**	**Expected Result**
1	Minder-ATV sends HTTP request below to DP searchServer to query status of delegation *https://dpUrlAddress/search?delegation{id}?&token={id}*	DP searchs revocation archieve and sends error message. To be defined when implemented.

ID	**TC_DP_12**	
Assertion(s)	TA_DP_2, TA_DP_1	
Test Purpose	Publish delegation key	
Pre-Test Conditions	Delegation Provider is accessible. Encrypted delegation key is needed.	
Step	**Test Activity**	**Expected Result**
1	Client sends the encyrpted delegation key to publication server. HTTP POST request to the following address: *https://dpUrlAddress/1/publish_key*	The service should return HTTP 201

ID	**TC_DP_13**
Assertion(s)	TA_DP_2, TA_DP_1
Test Purpose	Download delegation key
Pre-Test Conditions	Delegation Provider is accessible.

	Hash of public key is needed	
Step	**Test Activity**	**Expected Result**
1	Client sends the hash of its public key to publication server. Http GET request to the following address: *https://dpUrlAddress/1/download_key/{pkHash}?token={token}*	DP searches delegation key and returns it with HTTP 200 response.

8. References

LIGHTest. (2017, 04 28). *D2.3 – Requirements and Use Cases.* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20539492

LIGHTest. (2017, 02 28). *D2.14 – Reference Architecture.* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20534069

LIGHTest. (2017, 04 28). *D3.3 – DNS-based Publication of Trust Schemes.* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20539083

LIGHTest. (2017, 04 28). *D3.4 – Discovery of Trust Scheme Publication Authorities.* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20531317

LIGHTest. (2017, 04 28). *D4.3 – DNS-based Publication of Trust Translation Schemes.* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20531524

LIGHTest. (2017, 04 28). *D4.4 – Discovery of Trust Translation Authorities.* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20538593

LIGHTest. (2017, 04 28). *D5.2 – Conceptual Framework for Delegations (2).* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=20816230

LIGHTest. (2018, 09 03). *D8.3 – Conformance and Interoperability Result Report (1).* Fraunhofer Livelink: https://dms-prext.fraunhofer.de/livelink/livelink.exe?func=ll&objaction=overview&objid=21147476

9. Project Description

LIGHTest project to build a global trust infrastructure that enables electronic transactions in a wide variety of applications

An ever increasing number of transactions are conducted virtually over the Internet. How can you be sure that the person making the transaction is who they say they are? The EU-funded project LIGHTest addresses this issue by creating a global trust infrastructure. It will provide a solution that allows one to distinguish legitimate identities from frauds. This is key in being able to bring an efficiency of electronic transactions to a wide application field ranging from simple verification of electronic signatures, over eProcurement, eJustice, eHealth, and law enforcement, up to the verification of trust in sensors and devices in the Internet of Things.

Traditionally, we often knew our business partners personally, which meant that impersonation and fraud were uncommon. Whether regarding the single European market place or on a Global scale, there is an increasing amount of electronic transactions that are becoming a part of peoples everyday lives, where decisions on establishing who is on the other end of the transaction is important. Clearly, it is necessary to have assistance from authorities to certify trustworthy electronic identities. This has already been done. For example, the EC and Member States have legally binding electronic signatures. But how can we query such authorities in a secure manner? With the current lack of a worldwide standard for publishing and querying trust information, this would be a prohibitively complex leading to verifiers having to deal with a high number of formats and protocols.

The EU-funded LIGHTest project attempts to solve this problem by building a global trust infrastructure where arbitrary authorities can publish their trust information. Setting up a global infrastructure is an ambitious objective; however, given the already existing infrastructure, organization, governance and security standards of the Internet Domain Name System, it is with confidence that this is possible. The EC and Member States can use this to publish lists of qualified trust services, as business registrars and authorities can in health, law enforcement and justice. In the private sector, this can be used to establish trust in inter-banking, international trade, shipping, business reputation and credit rating. Companies, administrations, and citizens can then use LIGHTest open source software to easily query this trust information to verify trust in simple signed documents or multi-faceted complex transactions.

The three-year LIGHTest project starts on September 1st and has an estimated cost of almost 9 Million Euros. It is partially funded by the European Union's Horizon 2020 research and innovation programme under G.A. No. 700321. The LIGHTest consortium consists of 14 partners from 9 European countries and is coordinated by Fraunhofer-Gesellschaft. To reach out beyond Europe, LIGHTest attempts to build up a global community based on international standards and open source software.

The partners are ATOS (ES), Time Lex (BE), Technische Universität Graz (AT), EEMA (BE), G&D (DE), Danmarks Tekniske Universitet (DK), TUBITAK (TR), Universität Stuttgart (DE), Open Identity Exchange (GB), NLNet Labs (NL), CORREOS (ES), University of Piraeus Research Center (GR) and Globalsign (FI). The Fraunhofer IAO provides the vision and architecture for the project and is responsible for both, its management and the technical coordination.

The Fraunhofer IAO provides the vision and architecture for the project and is responsible for both, its management and the technical coordination.