A Lightweight Infrastructure for Global Heterogeneous Trust Management

Lightweight Infrastructure for Global Heterogeneous Trust management in support of an open Ecosystem of Stakeholders and Trust schemes
Reference Architecture of LIGHT\textsuperscript{est}
Trust Scheme Publication Authority (TSPA)

- Open Source Client Library and Server Tools (available on IAK Git) that aim to design
  - A conceptual framework to represent arbitrary trust schemes.
  - Trust schemes to be published/queried over DNS
  - The discovery of Trust Scheme Publication Authorities.
- Legal Toolbox, publicly available soon (M36 of the project),
  - Cross-Border Legal Compliance and Validity of this trust scheme publishing
Conceptual Framework for Trust Scheme of TSPA

- **DNS Name Server**
  - discovery of associated Trust Scheme and Trust Scheme Provider

- **Trust Scheme Provider**
  - signed trust list indicating issuer operates under the specific Trust Scheme (using existing standards on Trust Service Status Lists ETSI TS 119 612)
  - Tuple-based representation of Trust Scheme

---

[Diagram of Conceptual Framework for Trust Scheme of TSPA]

...
## Publication of Trust Schemes

<table>
<thead>
<tr>
<th>Type of Trust Scheme Publication</th>
<th>Example</th>
<th>Verifiable Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>ETSI_EN_319_401</td>
<td>Compliance of an entity to a trust scheme</td>
</tr>
<tr>
<td>Ordinal</td>
<td>LoA4.ISO29115</td>
<td>Compliance of an entity to an ordinal value of a trust scheme</td>
</tr>
<tr>
<td>Tuple-Based</td>
<td>{(authentication:2Factor), (identityProofing:inPerson)}</td>
<td>Requirements of a trust scheme</td>
</tr>
</tbody>
</table>
Tuple-Based Trust Scheme Representation

- Bottom-up modelling approach
  - Consolidation of existing trust schemes
  - Conceptualization of data model
  - Development of data model
  - Tuples (attribute_name, attribute_value)
- Modelling of Tuple-Based Trust Schemes

<table>
<thead>
<tr>
<th>Input Scheme 1</th>
<th>Input Scheme 2</th>
<th>Consolidation Result</th>
<th>Saturation $\Delta S$ (min $\Delta S$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO/IEC 29115</td>
<td>PCTF</td>
<td>Data Model v0.2</td>
<td>n.a.</td>
</tr>
<tr>
<td>Data Model v0.2</td>
<td>FIDO</td>
<td>Data Model v0.4</td>
<td>3</td>
</tr>
<tr>
<td>Data Model v0.4</td>
<td>QAA/AQAA, eIDAS</td>
<td>Data Model v0.6</td>
<td>9</td>
</tr>
<tr>
<td>Data Model v0.6</td>
<td>Chinese eSig Law</td>
<td>Data Model v0.6 (Data model of D3.1)</td>
<td>0</td>
</tr>
<tr>
<td>Data Model v0.6</td>
<td>Turkey eSig Law</td>
<td>Data Model v0.8</td>
<td>1</td>
</tr>
<tr>
<td>Data Model v0.8</td>
<td>MTF</td>
<td>Data Model</td>
<td>1</td>
</tr>
<tr>
<td>Data Model</td>
<td>Trust Scheme of Azerbaijan</td>
<td>Data Model</td>
<td>0</td>
</tr>
<tr>
<td>Data Model</td>
<td>UICC</td>
<td>Data Model</td>
<td>0</td>
</tr>
</tbody>
</table>

Wagner S. et al., 2019
Tuple-Based Trust Scheme Representation & Publication

- Data model
  - 27 concepts for Identity
  - 62 concepts for Credential
  - 9 concepts for Attributes
- 2 new constructs:
  - Authority Chain
  - Identity Provider
Tuple-Based Trust Scheme Representation&Publication

- Modelling of Tuple-Based Trust Schemes
  - Publication of Tuples of the generic Unified Data Model, e.g.
    ```xml
    <CredentialBindingUsingDigitalSignatures> true </CredentialBindingUsingDigitalSignatures>
    ```
  - Publication of Tuples-Based Trust Schemes
    - as part of the signed trust list
    - extra document with pointer from the trust list, e.g. `<AdditionalServiceInformation>`
DNS-based Trust Scheme Publication and Discovery

- Communication between components (DNS Name Server AND Trust Scheme Provider) for
  - Publishing Data using the TSPA: create, modify and delete Trust Schemes
  - Retrieving Data from the TSPA: querying process
Discovery of Trust Scheme Publication Authorities

- Example eIDAS Austria (with A-Trust as qualified trust service provider)

  - DNS query to discover trust scheme
    
    ;; QUESTION SECTION:
    _scheme._trust.a-trust.net. IN PTR
    
    ;; ANSWER SECTION:
    _scheme._trust.a-trust.net. IN PTR _scheme._trust.nrca-ds.at

  - DNS query to discover trust list
    
    ;; QUESTION SECTION:
    ;_scheme._trust.nrca-ds.at. IN URI
    
    ;; ANSWER SECTION:
    _scheme._trust.nrca-ds.at. IN URI https://www.nrca-ds.at/st/TSL-XML.xml
Discovery of Trust Scheme Publication Authorities

- Example eIDAS Austria (with D-Trust as qualified trust service provider) if
  - DNS query to discover certificate constraints
    
    ;; QUESTION SECTION:
    ;_scheme._trust.nrca-ds.at. IN SMIMEA
    ;; ANSWER SECTION:
    _scheme._trust.nrca-ds.at. IN SMIMEA <SMIMEA record data>
  
- <SMIMEA record data> example

  3 ; certificate usage domain issued cert
  0 ; selector: full certificate
  1 ; matching type SHA-256
c70cd54924d4c9cf ; certificate association data
  6ed20dc93c76aabb ...

Defined in RFC6698 & RFC7218
Trust Translation Authority (TTA)

- Open Source Client Library and Server Tools (available on IAK Git) that aim to design
  - A conceptual framework to represent arbitrary trust translation schemes.
  - Trust translation schemes to be published/queried over DNS
  - The discovery of Trust Translation Authorities.
- Legal Toolbox, publicly available soon (M36 of the project),
  - Cross-Border Legal Compliance and Validity of these trust translations publishing
TTA subcomponents
Discovery of Trust Translation Authorities

- how users (ATV) query TTA
  - find Trust Translations Lists
    1. QUESTION SECTION: Client/ATV to the TTA
    _translate._trust.loa4.eid.iso29115.org. IN URI
    _translate._trust.loa4.eid.iso29115.org. IN URI https://lightest.eu/ttl_LoA4iso29115_1.tpl
  
  - check validity of information
    1. QUESTION SECTION: Verifying authenticity
    _translate._trust.etimestamp.eidas.eu. IN SMIMEA
    _translate._trust.etimestamp.eidas.eu. IN SMIMEA <SMIMEA record data>

- Details:
  - ATM DNS Name Server w/DNSSEC Trust Translation Provider
  - Issue(DNSQuery=SchemeName)
  - LookupRR(SchemeName):...CertificateConstraints)
  - VerifySignature(Signature_TrustList,CertificateConstraints):Boolean

© LIGHTest Consortium
Trust Translation Scheme Representation

- Translations in TPL and XML formats
- A ternary list of (trustPolicy, sourceSchema, targetSchema).

\[
\begin{align*}
\text{translate_identity}(\text{EIDAS}, \text{FIDOUAF}_1, 5) :&= \\
&\quad \text{extract}(\text{EIDAS, schemename}, \text{eidas}), \\
&\quad \text{extract}(\text{FIDOUAF}_1, 5, \text{schemename}, \text{fidouaf}_1, 5), \\
&\quad \text{translate_qual}(\text{EIDAS, FIDOUAF}_1, 5).
\end{align*}
\]

\[
\begin{align*}
\text{translate_qual}(\text{EIDAS, FIDOUAF}_1, 5) :&= \\
&\quad \text{extract}(\text{EIDAS, eIdentity_level, qualified}), \\
&\quad \text{extract}(\text{FIDOUAF}_1, 5, \text{userVerification}, \text{"Fingerprint"}), \\
&\quad \text{extract}(\text{FIDOUAF}_1, 5, \text{userVerificationUp}, \text{"5"}).
\end{align*}
\]
Discovery of Trust Translation Lists

- Example: eIDAS eTimestamp
  - DNS query to discover trust translation lists
    - QUESTION SECTION: Client/ATV to the TTA
      ; _translate._trust.etimestamp.eidas.eu. IN URI
    - ANSWER SECTION: from the TTA
      - https://lightest.eu/ttl_qualifiedTimestampEidas1.tpl
      - https://lightest.eu/ttl_qualifiedTimestampEidasN.tpl
      - https://lightest.eu/ttl_qualifiedTimestampEidas1.xml
      - https://lightest.eu/ttl_qualifiedTimestampEidasN.xml
Verification of the Signed Translation Lists

Example eIDAS eTimestamp

- DNS query to discover certificate constraints

  ;; QUESTION SECTION:
  ;._translate._trust.etimestamp.eidas.eu IN SMIMEA
  ;; ANSWER SECTION:
  ;._translate._trust.etimestamp.eidas.eu IN SMIMEA <SMIMEA record data>

- <SMIMEA record data> example

  3 ; certificate usage domain issued cert
  0 ; selector: full certificate
  1 ; matching type SHA-256
  c70cd54924d4c9cf ; certificate association data
  6ed20dc93c76aabb …

Note: Defined in RFC6698 & RFC7218
Delegation Provider

- **Open Source Client Library and Server Tools** (available on IAK Git) that aim to design
  - A conceptual framework to represent delegations
  - Delegations to be published/queried
  - The discovery of Trust Translation Authorities.
- **Legal Toolbox**, publicly available soon (M36 of the project),
  - Cross-Border Legal Compliance and Validity of this delegations publishing
Design of a Conceptual Framework for Delegations

- Views on different projects and scientific publications
- Defines possible types of delegations (bilateral, substitution, delegation type)
- Data format defined
- Revocation of a delegation
  - Revocation with OCSP
  - Delegation Provider gets a delegation to sign the OCSP response
Design of Publication of Delegations

- Mandator
  - Creates delegation
  - Signs the delegation
  - Creates encryption key for the delegation
  - Encrypts generated key with Proxy’s public key
  - Uploads delegation and encryption key to delegation provider
Discovery of Delegations

- **Proxy**
  - Requests delegations
  - Provides public key

- **Delegation Provider**
  - Generates challenge
  - Sends challenge to proxy

- **Proxy**
  - Solves challenge
  - Sends result back

- **Delegation Provider**
  - Sends delegations to Proxy
How to Integrate and Test Components

- Sources can be obtained via IAIK GitLab at https://extgit.iaik.tugraz.at/LIGHTest/
- Each component uses/provides a REST API
  - TSPA to handle Trust Schemes that
    - Passes the information to the DNS server to create/update/delete entries
    - Stores the Trust Scheme information
  - TTA to handle Trust Translation Schemes that
    - Passes the information to the DNS server to create/update/delete entries
    - Stores the Trust Translation Scheme and Agreement information
  - DP
    - To create/update/delete entries
    - Stores delegation data
Integration and Conformance Testing of components in LIGHT*est

- Main objective
  - Render all LIGHT*est components mature and robust in order to reach TRL7.
  - Performs evaluations whether the products are in compliance with the defined specifications
- Iterative approach
  - 3 iterations are held
- Automated testing using Minder
How to Integrate and Test Components

- **Minder** Conformance and Interoperability Testbed is used for the testing architecture
- Implemented in e-SENS EU Project
- Ability to create-group-edit-execute test stories (or more formally test assertions converted to test cases) and inspect and publish reports and logs
- Minder Test Definition Language (MTDL, an extensible SCALA-based scripting language) including the use of external Java library dependencies
How to Integrate and Test Components

- **Minder** is compliant with GITB (Global e-Business Interoperability Test Bed methodologies).
- Focuses on methodologies and architectures that support e-business standards assessment and testing activities from early stages of business standards development from:
  - Implementation and
  - Implementation → deployment of large-scale solutions.
Integration and Conformance Testing of components in LIGHT\textsuperscript{est}

- Automate
- Testing Methodology is based on OASIS Test Assertion Model
Minder Testbed Applied Architecture

- The architecture & scenarios based on the design documentation is base on
  - Querying of Trust Scheme Membership
  - Querying of Trust Translation List
  - Discovering of Trust Delegation
  - Publishing of Trust Delegation Test Scenario

- **Minder Test Manager** is implemented to handle test case and suite execution
Conformance and Interoperability Testing Iterations

- TSPA
  - 18 Normative Statements:
  - 11 Test Assertions derived from normative statements
  - 20 Test Cases derived from assertions

- TTA
  - 15 Normative Statements
  - 15 Test Assertions
  - 25 Test cases

- DP
  - 13 Normative Statements
  - 15 Test Assertions
  - 18 Test cases
Conformance and Interoperability Testing in Summary

- Technical Infrastructure Setup – DNS with DNSSEC setup for the components
- Deployment and Integrating of LIGHTest components for testing
- Test Assertions and Test Cases extraction from:
  - Use cases = Integration Test
  - Requirements = Conformance and Interoperability Tests
- Test Executions and Report Generations
- Defect correction and Re-Execution of Tests automatically with minimum effort