E-LEGITIMATIONS
NÄMNDEN

Signature Activation Protocol for Federated Signing

Version 1.1 - 2018-08-03 - *Draft version*

ELN-0613-v1.1

Table of Contents

1. Introduction

This document specifies a **Signature Activation Protocol** (SAP) and its data elements for implementation of **Sole Control Assurance Level 2** (SCAL2) according the European standards prEN 419241 - Trustworthy Systems Supporting Server Signing - Part 1 and 2 (prEN 419 241-1 [RSIG-PP-1] and prEN 419 241-2 [RSIG-PP-2]).

The Signature Activation Protocol (SAP) defined in this document is used to exchange data between a signature service and a delegated authenticating authority such as a SAML Identity Provider. The function of the SAP is to authenticate the intent of a signer to sign a particular document, or collection of documents, through exchange of the following data elements.

- Signature Activation Data (SAD) - Signed data, asserting the signer's agreement to sign specific data.
- SADRequest - Request for a SAD.

The SAP specified in this document is specifically designed to be used with a signing service operating in accordance with the federated signing specification [ELN-0609].

1.1. Requirement key words

The key words **MUST**, **MUST NOT**, **REQUIRED**, **SHALL**, **SHALL NOT**, **SHOULD**, **SHOULD NOT**, **RECOMMENDED**, **MAY**, and **OPTIONAL** are to be interpreted as described in [RFC2119].

These keywords are capitalized when used to unambiguously specify requirements over protocol features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

1.2. XML name space references

The prefix **sap:** stands for the Signature Activation Protocol XML Schema namespace `http://id.elegnamnden.se/csig/1.1/sap/ns` (https://elegnamnden.github.io/schemas/csig/1.1/EidCsigSAP-1.1.xsd).

The prefix **saml2:** stands for the OASIS SAML 2 Assertion Schema namespace `urn:oasis:names:tc:SAML:2.0:assertion`.

The prefix **saml2p:** stands for the OASIS SAML 2 Protocol Schema namespace `urn:oasis:names:tc:SAML:2.0:protocol`.

1.3. Structure

This specification uses the following typographical conventions in text: `<Eid2Element>`, `<ns:ForeignElement>`, `Attribute`, **Datatype**, `OtherCode`.

2. Signature Activation Protocol

2.1. Scope

The scope of the Signature Activation Protocol (SAP) is to support request for and delivery of the Signature Activation Data (SAD) to the Signature Activation Module (SAM). The SAM is a tamper resistant module inside the Remote Signing Service which validates the SAD in order to ensure that:

- the signer is properly authenticated,
- the signer agrees to sign the data to be signed, and,
- the correct signing key for this signer and this instance of signing is properly identified.

The federated signing model does not use pre-assigned signing keys. Instead, a new signing key is generated for each sign request and then permanently deleted. This particular use-case is recognised by prEN 419 241-1 [RSIG-PP-1] and prEN 419 241-2 [RSIG-PP-2], which under these conditions allows the signature key reference to be implicit and derived from the signer's identity. For the present implementation of the SAP the following data is included in the SAD:

- the signer's identity,
- information about how the signer was authenticated and by whom, and,
- reference to the data to be signed.

This implements the scenario where the Identity Provider is the sole entity which can verify the signer's private credentials via the SIC (Signer’s Interaction Component). This instance of authentication is used by the Identity Provider to generate the SAD in accordance with section 5.10 of [RSIG-PP-1].

2.2. Data exchange

This document specifies exchange of two data elements:

- `SADRequest`
- `SAD`

The `SADRequest` SHALL have the format defined in section 3.1. When a Remote Signing Service request a SAD from the Identity Provider, it MUST include the `SADRequest` element as an request extension by including it as a child element to a `<saml2p:Extensions>` element in the `<saml2p:AuthnRequest>`.

When an Identity Provider returns a SAD, as defined in section 3.2, in a SAML Assertion, it MUST be included as a single string value of a `sad` attribute identified by the attribute name `urn:oid:1.2.752.201.3.12` as defined in the attribute specification [ELN-0604].

3. Data elements

The SAD requested in the SAP binds the documents to be signed to the intent by the signer to sign. This is accomplished by the interaction of a number of independent information attributes and elements as follows:

- **Sign request ID**. Identifies the sign request message for signing specific documents. This sign request is sent to the signing service from the service provider requesting the signature. The sign request bound by this identifier contains all detailed data about what is being signed.
- **Sign message**. A description of what is being signed that is passed from the service provider requesting signing to the Identity Provider, via the signing service. The sign message is included in the sign request as well as in the SAML authentication request sent to the Identity Provider.
- **LoA**. The level of assurance declaration asserting the level of security used to authenticate the user and asserting that the user read and accepted the sign message and approved to sign the document/s.
- **Number of documents to sign**. Ensures that the user is aware whether more than one document is being signed. This allows adaptations of the signing UI displayed by the Identity Provider.
- **Identity of the signer**. Allows verification that the signature is bound to the appropriate signer.
- **SAD Request ID**. Unique identifier for the `SADRequest` element. This identifier is later included in the SAD in order to accomplish a binding between the request and the issued SAD.

The SAD request and the SAD specified in this section specifies the data that needs to be exchanged in addition to other protocol elements specified by SAML as well as the federated signing specification [ELN-0609].

3.1. SADRequest

3.1.1 Syntax

The SAD Request is provided in a `<sap:SADRequest>` element. The element has the following elements and attributes:

`<RequesterID>` [Required]

> Specifies the SAML entityID of the requesting entity. The value for this element should be the same identifier as given in the `<saml2:Issuer>` element of the `<saml2p:AuthnRequest>` that encapsulates the `<sap:SADRequest>` extension.

`<SignRequestID>` [Required]

> Specifies the value of the `RequestID` attribute of the associated `SignRequest`.

`<DocCount>` [Required]

> The number of requested signatures in the associated sign request.

`<RequestedVersion>` [Optional Default="1.0"]

> The requested version of the SAD.

`<RequestParams>` [Optional]

> Optional parameters provided as name-value pairs. This specification does not define any parameters. The use of parameters may be defined in profiles of this specification or may be negotiated by other means between a remote signing service and an Identity Provider.

`ID` [Required]

> Attribute holding an unique identifier for the `SADRequest`.

The following schema fragment defines the `<sap:SADRequest>` element:

```
<xs:element name="SADRequest" type="sap:SADRequestType" />

<xs:complexType name="SADRequestType">
  <xs:sequence>
    <xs:element name="RequesterID" type="xs:string" />
    <xs:element name="SignRequestID" type="xs:string" />
    <xs:element name="DocCount" type="xs:int" />
    <xs:element name="RequestedVersion" type="xs:string" minOccurs="0" default="1.0" />
    <xs:element minOccurs="0" name="RequestParams">
      <xs:complexType>
        <xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="0" name="Parameter" type="sap:ParameterType" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:ID" use="required" />
</xs:complexType>

<xs:complexType name="ParameterType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="name" type="xs:string" use="required" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

3.1.2 Example

```
<sap:SADRequest ID="_a74a068d0548a919e503e5f9ef901851" xmlns:sap="http://id.elegnamnden.se/csig/1.1/sap/ns">
  <sap:RequesterID>http://www.example.com/sigservice</sap:RequesterID>
  <sap:SignRequestID>f6e7d061a23293b0053dc7b038a04dad</sap:SignRequestID>
  <sap:DocCount>1</sap:DocCount>
  <sap:RequestedVersion>1.0</sap:RequestedVersion>
  <sap:RequestParams>
    <sap:Parameter name="ParamName">paramValue</sap:Parameter>
  </sap:RequestParams>
</sap:SADRequest>
```

Example of a SADRequest.

3.2. Signature Activation Data

3.2.1. SAD JSON Web Token

This section specifies a JSON Web Token (JWT) in accordance with [RFC7519] as the SAD container, binding the data as defined in section 2.1.

The SAD JWT MUST have the form of a signed JWS (JSON Web Signature).

3.2.1.1. Registered JWT claims

The data signed by the SAD JWT is carried in the JWS payload in the form of JWT claims using registered claim names (as specified in [RFC7519]) in addition to one private claim name (`seElnSadext`) specified in section 3.2.1.2. The following table defines the use of registered claims.

name	Content
sub	Subject - holds the attribute value of the signer's unique identifier.
aud	Audience - holds the entityID of the Signature Service which is the legitimate recipient of this SAD. This value corresponds to the `<sap:RequesterID>` element of the SAD request.
iss	Issuer - holds the entityID of the IdP that generated this SAD.
exp	Expiry - specifies the time when this SAD is no longer valid (epoch time/seconds since 1970-01-01).
iat	Issued At - specifies the time when this SAD was issued (epoch time/seconds since 1970-01-01).
jti	Unique identifier of this SAD.

3.2.1.2. SAD Extension claim

A private claim name is defined in this specification which extends the registered claims with additional SAD data:

> `seElnSadext`

The claim identified by this name has the value of a JSON object holding name-value pairs in accordance with the following table:

Name	Type	Content
ver	String	The version of this claim, default 1.0 (Optional).
irt	String	In Response To - holds the identifier of the SAD request (`ID` attribute) that was used to request this SAD.
attr	String	Attribute - holds the URI identifier of the attribute specifying the users unique identifier value.
loa	String	LevelOfAssurance - holds the URI identifier of the level of assurance (LoA) used to authenticate the signer.
reqid	String	RequestID - holds the ID of the sign request associated with this SAD.
docs	Integer	Specifies the number of documents to be signed in the associated sign request.

3.2.2. Example

The following example illustrates a claim binding the following claim values:

Registered Claims

Name	Value
sub	`196302052383`
aud	`http://www.example.com/sigservice`

Name	Value
iss	`https://idp.svelegtest.se/idp`
exp	`1516195657` (2018-01-17 13:27:37 GMT)
iat	`1516195357` (2018-01-17 13:22:37 GMT)
jti	`d4073fc74b1b9199`

seElnSadext Claim

Name	Value
ver	`1.0`
irt	`_a74a068d0548a919e503e5f9ef901851`
attr	`urn:oid:1.2.752.29.4.13`
loa	`http://id.elegnamnden.se/loa/1.0/loa3-sigmessage`
reqid	`f6e7d061a23293b0053dc7b038a04dad`
docs	`1`

JWS Header

The Header of the JWS specifies that it is a JWT by the "typ" parameter and the signature algoritm through the "alg" parameter. In this example the header is `{"typ":"JWT","alg":"RS256"}`. The Base64 URL-encoded header is:

> eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9

JWS Payload

The JWS payload holding the JWT claims is represented by the following JSON object:

```
{
    "sub" : "196302052383",
    "aud" : "http://www.example.com/sigservice",
    "iss" : "https://idp.svelegtest.se/idp",
    "exp" : 1516195657,
    "iat" : 1516195357,
    "jti" : "d4073fc74b1b9199",
    "seElnSadext" : {
        "ver" : "1.0",
        "irt" : "_a74a068d0548a919e503e5f9ef901851",
        "attr" : "urn:oid:1.2.752.29.4.13",
        "loa" : "http://id.elegnamnden.se/loa/1.0/loa3-sigmessage",
        "reqid" : "f6e7d061a23293b0053dc7b038a04dad",
        "docs" : 1
    }
}
```

This payload is represented by the following Base64 URL-encoded string:

eyJzdWIiOiIxOTYzMDIwNTIzODMiLCJhdWQiOiJodHRwOi8vd3d3LmV4YW1wbGUuY29tL3NpZ3NlcnZpY2UiLCJpc3MiOi
JodHRwczovL2lkcC5zdmVsZWd0ZXN0LnNlL2lkcCIsImV4cCI6MTUyMDUwNDExMCwiaWF0IjoxNTIwNTAzODEwLCJqdG
kiOiIyeGlUdEZOSE5iTWpweE1yUTh0RWZHY3AiLCJzZUVsblNhZGV4dCI6eyJ2ZXIiOiIxLjAiLCJpcnQiOiJfYTc0YTA2OGQw
NTQ4YTkxOWU1MDNlNWY5ZWY5MDE4NTEiLCJhdHRyIjoidXJuOm9pZDoxLjIuNzUyLjI5LjQuMTMiLCJsb2EiOiJodHRwO
i8vaWQuZWxlZ25hbW5kZW4uc2UvbG9hLzEuMC9sb2EzLXNpZ21lc3NhZ2UiLCJyZXFpZCI6ImY2ZTdkMDYxYTIzMjkzYjA
wNTNkYzdiMDM4YTA0ZGFkIiwiZG9jcyI6MX19

JWT

The complete SAD JWT including signature:

eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIxOTYzMDIwNTIzODMiLCJhdWQiOiJodHRwOi8vd3d3LmV4YW1w
bGUuY29tL3NpZ3NlcnZpY2UiLCJpc3MiOiJodHRwczovL2lkcC5zdmVsZWd0ZXN0LnNlL2lkcCIsImV4cCI6MTUyMDUwND
ExMCwiaWF0IjoxNTIwNTAzODEwLCJqdGkiOiIyeGlUdEZOSE5iTWpweE1yUTh0RWZHY3AiLCJzZUVsblNhZGV4dCI6eyJ2
ZXIiOiIxLjAiLCJpcnQiOiJfYTc0YTA2OGQwNTQ4YTkxOWU1MDNlNWY5ZWY5MDE4NTEiLCJhdHRyIjoidXJuOm9pZDoxLjI
uNzUyLjI5LjQuMTMiLCJsb2EiOiJodHRwOi8vaWQuZWxlZ25hbW5kZW4uc2UvbG9hLzEuMC9sb2EzLXNpZ21lc3NhZ2UiL
CJyZXFpZCI6ImY2ZTdkMDYxYTIzMjkzYjAwNTNkYzdiMDM4YTA0ZGFkIiwiZG9jcyI6MX19.jL0ccMQJmGx3UMDrNkVhFU
F5iIHiLD1r-
roSJsSzyUsEfEDaIBdbmjw7IpBILb2j69YTA_2z3WNBRkzTvHbYsDYucShMPQB7hbk41_oItKqTI038Y3FQXyExaNDZ7sHY
K4HSQQc53JPBplu1iMsjm9VTTI9VVrWahj-1-aFbC8LmdhVNFenLuJrs_tmPIGUa_mQC61-
46uKxs2Xq4NFpwdUToApAPawzIg2DktztotBVdJFvJj8nrQ9La8quHIdXaGyg5i2jG7YPA-
t3Khuooie_Ja0RTAte6bvTC4YGZhj-hQUsZ4M3CDrwdyRJ5zF_HXSmviLlwvjEDekFEP_j3g

3.2.3. Verification of a SAD

The recipient of a requested SAD MUST verify it as part of the SAML response processing by asserting the following:

- That the signature of the SAD JWT verifies correctly using the signature certificate of the issuing Identity Provider (found in the Identity Provider metadata).
- That the version of the SAD (`seElnSadext.ver`) matches the `<sap:RequestedVersion>` element of the `<sap:SADRequest>`.
- That the audience (`aud`) matches the entityID of the recipient, i.e., matches the `<sap:RequesterID>` element from the `<sap:SADRequest>`.
- That the issuer (`iss`) value matches the issuer entityID of the assertion containing the SAD (*).
- That the SAD is valid by checking the expiry (`exp`) and issued-at (`iat`) values (allowing for a reasonable clock skew).
- That the in-response-to (`seElnSadExt.irt`) value matches that `ID` of the corresponding `<sap:SADRequest>`.
- That the subject (`sub`) value is also represented in the SAML assertion as an attribute having the name given by the `seElnSadExt.attr` field.
- That the level of assurance (`seElnSadEx.loa`) value matches the value given in the `<saml2:AuthnContextClassRef>` element of the assertion.
- That the request ID (`seElnSadEx.reqid`) value matches the ID for the sign request (which is passed in the `<sap:SignRequestID>` element of the `<sap:SADRequest>`).
- That the number of documents specified in the SAD (`seElnSadEx.docs`) matches the `<sap:DocCount>` element of the `<sap:SADRequest>`.

If any of the above verification steps fail, the Signature Service MUST reject the assertion.

> * - In the case where a Signature Service communicates with a Proxy Identity Provider that forwards requests to an *authenticating* Identity Provider that issues a SAD, the `iss`-value of the SAD will differ from the issuer of the assertion that is received by the Signature Service. In these cases the Signature Service should compare the `iss`-value with the value found in the `<saml2:AuthenticatingAuthority>` element of the assertion, or with relevant local policy and out-of-band configuration data.

4. Schemas

The following XML schema defines the `http://id.elegnamnden.se/csig/1.1/sap/ns` name space:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
    targetNamespace="http://id.elegnamnden.se/csig/1.1/sap/ns"
    xmlns:sap="http://id.elegnamnden.se/csig/1.1/sap/ns">

  <xs:annotation>
    <xs:documentation>
      Schema location URL: https://elegnamnden.github.io/schemas/csig/1.1/EidCsigSAP-1.1.xsd
    </xs:documentation>
  </xs:annotation>

  <xs:element name="SADRequest" type="sap:SADRequestType" />

  <xs:complexType name="SADRequestType">
    <xs:sequence>
      <xs:element name="RequesterID" type="xs:string" />
      <xs:element name="SignRequestID" type="xs:string" />
      <xs:element name="DocCount" type="xs:int" />
      <xs:element name="RequestedVersion" type="xs:string" minOccurs="0" default="1.0" />
      <xs:element minOccurs="0" name="RequestParams">
        <xs:complexType>
          <xs:sequence>
            <xs:element maxOccurs="unbounded" minOccurs="0" name="Parameter" type="sap:ParameterType" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ID" type="xs:ID" use="required" />
  </xs:complexType>

  <xs:complexType name="ParameterType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="name" type="xs:string" use="required" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>
```

5. Normative References

[RFC2119]

> Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, March 1997.

[RFC7519]

> Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015.

[ELN-0604]

> Attribute Specification for the Swedish eID Framework.

[ELN-0609]

> DSS Extension for Federated Central Signing Services.

[RSIG-PP-1]

> European Standard prEN 419241-1 - Trustworthy Systems Supporting Server Signing - Part 1: General System Security Requirements

[RSIG-PP-2]

> European Standard prEN 419241-2 - Trustworthy Systems Supporting Server Signing - Part 2: Protection profile for QSCD for Server Signing

6. Changes between versions

Changes between version 1.0 and 1.1:

- The `RequestedVersion` element of the `SADRequestType` is now marked as optional in the schema definition.