Deployment Profile for the Swedish eID Framework

ELN-0602-v1.4
Version 1.4
2016-08-30

1 Introduction

This profile specifies behavior and options that deployments of the SAML V2.0 Web Browser SSO Profile [SAML2Prof] are required or permitted to rely on. The profile extends Interoperable SAML 2.0 Web Browser SSO Deployment Profile [SAML2Int] with requirements specific for the Swedish eID Framework and specifies deployment details that are not covered in [SAML2Int].

Readers should be familiar with all relevant reference documents, and any requirements stated are not repeated unless where deemed necessary to clarify or highlight a certain issue.

This profile, like [SAML2Int], addresses the content, exchange, and processing of SAML messages, but also specifies some deployment details that go beyond that scope, such as required metadata elements.

Any SAML features specified in referenced SAML documents that are optional are out of scope of this profile, unless explicitly specified by this profile.

This profile does not handle requirements regarding algorithms and different versions of underlying security mechanisms. This information is distributed by the federation operator in other channels.

1.1 Requirements Notation

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

The use of SHOULD, SHOULD NOT, and RECOMMENDED reflects broad consensus on deployment practices intended to foster both interoperability and guarantees of security and confidentiality needed to satisfy the requirements of many organizations that engage in the use of federated identity. Deviating may limit a deployment's ability to technically interoperate without additional negotiation, and should be undertaken with caution.

1.2 References to SAML 2.0 Standards and Profiles

When referring to elements from the SAML 2.0 core specification [SAML2Core], the following syntax is used:

- `<saml2p:Protocolelement>` – for elements from the SAML 2.0 Protocol namespace.
- `<saml2:Assertionelement>` – for elements from the SAML 2.0 Assertion namespace.

When referring to elements from the SAML 2.0 metadata specifications, the following syntax is used:

- `<md:Metadataelement>` – for elements defined in [SAML2Meta].
- `<mdui:Element>` – for elements defined in [SAML2MetaUI].
- `<mdattr:Element>` – for elements defined in [SAML2MetaAttr].

When referring to elements from the Identity Provider Discovery Service Protocol and Profile [IdPDisco], the following syntax is used:

- `<idpdisc:DiscoveryResponse>`

When referring to elements from the W3C XML Signature namespace (http://www.w3.org/2000/09/xmldsig#) the following syntax is used:

- `<ds:Signature>`

2 Metadata and Trust Management

Identity Providers and Service Providers that are part of the federation for Swedish eID MUST provide a SAML 2.0 Metadata document representing its entity. Provided metadata MUST conform to [SAML2Int] as well as the SAML V2.0 Metadata Interoperability Profile Version 1.0 [MetaIOP].

2.1 Requirements for Metadata Content

2.1.1 Generic

All services that are represented in the Metadata SHALL include a `<md:Organization>` element with mandatory child elements, which includes at least one of each of the elements `<md:OrganizationName>`, `<md:OrganizationDisplayName>` and `<md:OrganizationURL>`.

The `<md:OrganizationName>` element SHALL hold a registered name of the organization, which matches the agreement with the federation operator.

The `<md:OrganizationDisplayName>` element SHALL contain a display name of the organization and SHALL NOT contain a service name that is unrelated to the name of the organization.

All services represented in the metadata SHALL include RSA public keys in the form of a certificate, which supports both signature validation and encryption. The same public key MAY support both signature validation and encryption, indicated by an absent `"use"` attribute.

2.1.2 Service Providers

The `<mdattr:EntityAttributes>` element of a Service Provider's entity descriptor SHOULD contain one entity category attribute [EntCat] that holds at least one attribute value representing a service entity category as defined in [Eid2EntCat], identifying the Service Provider needs in relation to identity services.

The example below illustrates how an entity declares the service entity category identifier **http://id.elegnamnden.se/ec/1.0/loa3-pnr** in its metadata.

```
...
<md:Extensions>
  <mdattr:EntityAttributes xmlns:mdattr="urn:oasis:names:tc:SAML:metadata:attribute">
    <saml2:Attribute Name="http://macedir.org/entity-category"
                     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                     xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
      <saml2:AttributeValue xsi:type="xs:string">http://id.elegnamnden.se/ec/1.0/loa3-pnr</saml2:AttributeValue>
    </saml2:Attribute>
  </mdattr:EntityAttributes>
</md:Extensions>
...
```

Any needs for particular attributes from Identify Providers, when present, MUST be expressed through present service entity categories in combination with `<md:RequestedAttribute>` elements in the Service Provider metadata. The `<md:RequestedAttribute>` elements in the Service Provider metadata, when present, hold a list of requested and/or required attributes. This list of attributes MUST be interpreted in the context of present service entity categories defined in [EidEntCat].

Metadata for a Service Provider SHALL contain an `<mdui:UIInfo>` extension, extending the `<md:SPSSODescriptor>` element. This `<mdui:UIInfo>` element SHALL at least contain a `<mdui:DisplayName>` element with the language attribute `"sv"` (Swedish), representing the Service Provider

name that has been approved by the federation operator. The <mdui:UIInfo> element SHALL also contain a reference to a logotype image (<mdui:Logo>) and SHOULD contain a <mdui:Description> element with the language attribute "sv" (Swedish).

It is RECOMMENDED that the above elements represented in Swedish also be represented with the language attribute "en" (English).

A Service Provider MAY sign authentication request messages sent to Identity Providers. A Service Provider that signs authentication requests messages MAY also ensure that a receiving Identity Provider will only accept valid signed requests from this Service Provider by assigning the AuthnRequestsSigned attribute of the <md:SPSSODescriptor> to a value of "true".

Section E7, "Metadata for Agreeing to Sign Authentication Requests", of [SAML v2.0 Errata 05] specifies the following concerning the AuthnRequestsSigned attribute:

> Optional attribute that indicates whether the <saml2p:AuthnRequest> messages sent by this service provider will be signed. If omitted, the value is assumed to be false. A value of false (or omission of this attribute) does not imply that the service provider will never sign its requests or that a signed request should be considered an error. However, an identity provider that receives an unsigned <saml2p:AuthnRequest> message from a service provider whose metadata contains this attribute with a value of true MUST return a SAML error response and MUST NOT fulfill the request.

Furthermore, a Service Provider MAY require assertions that are issued to it, to be signed. This is done by assigning the WantAssertionsSigned attribute of the <md:SPSSODescriptor> to a value of "true".

Note that the response message that carries the assertion will always be signed, so the Service Provider should only require signed assertions in case that it wants to preserve the proof of authenticity of an assertion separate from the response.

2.1.3 Identity Providers

The <mdattr:EntityAttributes> element of an Identity Provider's entity descriptor SHOULD contain one entity category attribute [EntCat] that holds at least one attribute value representing a service entity category as defined in [EidEntCat], defining the Identity Provider ability to deliver assertions.

The <mdattr:EntityAttributes> element of an Identity Provider's metadata SHALL contain an attribute according to [SAML2IAP] with Name="urn:oasis:names:tc:SAML:attribute:assurance-certification" holding at least one attribute value identifying a Level of Assurance (LoA) level for which the Identity Provider has been approved and where the value is one of the identifiers defined in section 3.1.1 of [EidRegistry] and whose meaning are defined in [EidTillit].

Metadata for an Identity Provider SHALL contain an <mdui:UIInfo> extension, extending the <md:IDPSSODescriptor> element. This <mdui:UIInfo> element SHALL at least contain a <mdui:DisplayName> element with the language attribute "sv" (Swedish), representing the Identity Provider service name that has been approved by the federation operator. The <mdui:UIInfo> element SHALL also contain a reference to a logotype image (<mdui:Logo>) and SHOULD contain a <mdui:Description> element with the language attribute "sv" (Swedish).

It is RECOMMENDED that the above elements represented in Swedish also be represented with the language attribute "en" (English).

An Identity Provider MAY require authentication request messages to be signed. This is indicated by assigning the `WantAuthnRequestsSigned` attribute of the `<md:IDPSSPDescriptor>` element to a value of `"true"`. See further section E7, “Metadata for Agreeing to Sign Authentication Requests”, of [SAML v2.0 Errata 05].

2.1.4 Signature Service

The Signature Service within the framework for Swedish eID is a Service Provider with specific requirements concerning its representation in metadata. Its entry in metadata SHALL contain an `<mdui:UIInfo>` element, extending the `<md:SPSSODescriptor>` element. This `<mdui:UIInfo>` element SHALL at least contain a `<mdui:DisplayName>` element with the language attribute `"sv"` (Swedish), representing the signature service that has been approved by the federation operator.

The `<mdui:UIInfo>` element SHALL also contain a reference to a logotype image (`<mdui:Logo>`) and at least contain one `<mdui:Description>` element with the language attribute `"sv"` (Swedish), providing a description of the service according to requirements provided by the federation operator.

It is RECOMMENDED that the above elements represented in Swedish also be represented with the language attribute `"en"` (English).

The `<mdattr:EntityAttributes>` element of a Signature Service SP entity descriptor SHALL include the service type entity category identifier **http://id.elegnamnden.se/st/1.0/sigservice** [EidEntCat] as a value to the entity category attribute [EntCat].

```
...
<mdattr:EntityAttributes xmlns:mdattr="urn:oasis:names:tc:SAML:metadata:attribute">
  <saml2:Attribute Name="http://macedir.org/entity-category"
                   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                   xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
    <saml2:AttributeValue xsi:type="xs:string">http://id.elegnamnden.se/ec/1.0/loa3-pnr</saml2:AttributeValue>
    <saml2:AttributeValue xsi:type="xs:string">http://id.elegnamnden.se/st/1.0/sigservice</saml2:AttributeValue>
  </saml2:Attribute>
</mdattr:EntityAttributes>
...
```

Entity attributes for a Signature Service SP.

A Signature Service MUST assign the `AuthnRequestsSigned` attribute of the `<md:SPSSODescriptor>` element to `"true"`. This requirement ensures that the Signature Service always signs its authentication requests in order for the request to be accepted by the Identity Provider. The federation operator will enforce that all Service Providers that operate as Signature Services have this attribute set.

3 Name Identifiers

Identity Providers and Service Providers MUST support both the `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` and the `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` name identifier formats as specified in [SAML2Core].

Identity Providers SHALL default to use the `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent` name identifier format in cases where a Service Provider has not specified the name identifier to use (via the `<md:NameIDFormat>` element of the Service Provider metadata entry, or via the `Format` attribute of the `<saml2p:NameIDPolicy>` element of the authentication request message).

4 Attributes

Attribute specifications for the Swedish eID Framework is defined in [EidAttributes].

The content of `<saml2:AttributeValue>` elements exchanged via any SAML 2.0 messages or assertions SHOULD be limited to a single child text node.

For requirements regarding attribute inclusion in SAML assertions, see section 6.2.1, “Attribute Release Rules”, below.

5 Authentication Requests

5.1 Discovery

The federation for Swedish eID uses a central discovery service as specified in Identity Provider Discovery Service Protocol Profile [IdPDisco]. A Service Provider is not obliged to use the central discovery service and MAY instead implement discovery using an integrated technique as described in [EidDisco].

A Service Provider SHOULD use either the central discovery service or the integrated discovery techniques as described in [EidDisco].

Service Providers making use of the central discovery service MUST be able to handle empty responses for the cases where no Identity Provider was chosen. In these cases an error message should be displayed for the end user.

5.2 Binding and Security Requirements

The endpoints, at which an Identity Provider receives a `<saml2p:AuthnRequest>` message, and all subsequent exchanges with the user agent, MUST be protected by TLS/SSL ([SAML2Int] specifies SHOULD).

[SAML2Int] specifies that a `<saml2p:AuthnRequest>` message MUST be communicated to the Identity Provider using the HTTP-REDIRECT binding. This profile will also allow the usage of the HTTP-POST binding for sending `<saml2p:AuthnRequest>` messages (see section 3.5 of [SAML2Bind]).

An Identity Provider that requires `<saml2p:AuthnRequest>` messages to be signed MUST not accept messages that are not signed, or where the verification of the signature fails. In these cases the Identity Provider MUST respond with an error.

An Identity Provider that itself does not require authentication messages to be signed MUST still accept and verify signed request messages from Service Providers that indicate, in their metadata, that they sign request messages (see 2.1.2 above). If this signature verification fails, the Identity Provider MUST return a SAML error response and MUST NOT fulfill the request.

An Identity Provider that receives a request message that is not signed from a Service Provider that has indicated, in its metadata, that it will only send signed request messages (see 2.1.2 above) MUST respond with an error.

The signature for authentication request messages is applied differently depending on the binding. The HTTP-REDIRECT binding requires the signature to be applied to the URL-encoded value rather than being placed within the XML-message (see section 3.4.4.1 of [SAML2Bind]). For the HTTP-POST binding the `<saml2p:AuthnRequest>` element MUST be signed using a `<ds:Signature>` element within the `<saml2:AuthnRequest>`.

5.3 Message Content

[SAML2Int] specifies that a `<saml2p:AuthnRequest>` message SHOULD contain an `AssertionConsumerServiceURL` attribute identifying the desired response location. The Service Provider MUST NOT use any other values for this attribute than those listed in its metadata record as `<md:AssertionConsumerService>` elements for the HTTP-POST binding (see section 4.1.6 of [SAML2Prof]).

The `Destination` attribute of the `<saml2p:AuthnRequest>` message MUST contain the URL to which the Service Provider has instructed the user agent to deliver the request. This is useful to prevent malicious forwarding of signed requests from being accepted by unintended Identity Providers.

A Service Provider SHOULD explicitly specify one requested authentication context element (`<saml2p:RequestedAuthnContext>`), containing one or more `<saml2:AuthnContextClassRef>` elements that each contains an authentication context URI[1] representing a defined Level of Assurance under which the authentication process should be performed. A present `<saml2p:RequestedAuthnContext>` element MUST specify exact matching by means of either an absent `Comparison` attribute or a `Comparison` attribute with the value set to `exact`. This means that the Identity Provider is forced to return an assertion with exactly one of the requested `<saml2:AuthnContextClassRef>` in the request as the declared `<saml2:AuthnContext>`, or return an error response. If the Service Provider requires the Identity Provider to return specifically one out of a selection of acceptable authentication context URIs, then all of these URIs MUST be included in the request.

The requested authentication context SHOULD be consistent with at least one of the service entity categories [EidEntCat] declared in the Service Provider's metadata entry. See further section 5.4.4 below.

[1] See section 3.1.1 of [EidRegistry].

```
...
<saml2p:RequestedAuthnContext Comparison="exact">
  <saml2:AuthnContextClassRef>http://id.elegnamnden.se/loa/1.0/loa3</saml2:AuthnContextClassRef>
</saml2p:RequestedAuthnContext>
...
```

Example of how an Authentication Context URI identifier representing a requested Level of Assurance is included in an authentication request message.

```
<saml2p:RequestedAuthnContext Comparison="exact">
  <saml2:AuthnContextClassRef>http://id.elegnamnden.se/loa/1.0/loa3</saml2:AuthnContextClassRef>
  <saml2:AuthnContextClassRef>http://id.elegnamnden.se/loa/1.0/eidas-nf-sub</saml2:AuthnContextClassRef>
</saml2p:RequestedAuthnContext>
...
```

Example of how several Authentication Context URIs are included in an authentication request message. In this case, the Service Provider states that it requests the authentication to be performed according to either the LoA3 URI defined within the Swedish eID Framework or the substantial level for notified eIDs defined within the eIDAS Framework.

Identity Providers conformant with this profile MUST support the `ForceAuthn` and `IsPassive` attributes received in `<saml2p:AuthnRequest>` messages.

Service Providers SHOULD include the `ForceAuthn` attribute in all `<saml2p:AuthnRequest>` messages and explicitly set its value to `true` or `false`, and not rely on its default value. The reason for this is to avoid accidental SSO.

5.4 Processing Requirements

5.4.1 Validation of Destination

An Identity Provider receiving a `<saml2p:AuthnRequest>` message MUST verify that the `Destination` attribute is present, and that it is consistent with URLs configured in the Identity Provider's metadata.

5.4.2 Validation of Assertion Consumer Addresses

If the `AssertionConsumerServiceURL` attribute is present in the `<saml2p:AuthnRequest>` message, its value MUST be verified to be consistent with one of the `<md:AssertionConsumerService>` elements having the HTTP-POST binding found in the Service Provider's metadata entry. If this is not the case, the request must be rejected.

If the attribute is not present in the `<saml2p:AuthnRequest>` message, the Identity Provider MUST obtain the desired response location from the Service Provider's metadata entry. This location is found in an `<md:AssertionConsumerService>` element with HTTP-POST binding that is marked as default (has the `isDefault` attribute set), or if no element has the `isDefault` attribute set, the one with the lowest index value (see section 2.4.4.1 of [SAML2Meta]).

Section 8.2 of [SAML2Int] specifies how comparisons between the `AssertionConsumerServiceURL` value and the values found in the Service Provider's metadata should be performed.

5.4.3 Identity Provider User Interface

Where the requirements for user interfaces defined for the federation requires presentation of information elements related to the Service Provider, these information elements MUST be obtained from the `<mdui:UIInfo>` element in the Service Provider's metadata entry. Implementers of this profile MUST be ca-

pable of handling display information stored in the `<mdui:DisplayName>`, `<mdui:Logo>` and the `<mdui:Description>` elements.

5.4.4 Authentication Context and Level of Assurance Handling

This framework defines a number of authentication context identifiers (URI), where each such identifier specifies a defined Level of Assertion and may define specific requirements on the authentication process. There can be multiple authentication context URIs representing the same Level of Assertion, but one authentication context URI always identifies one defined Level of Assurance. For example, requests for authentication from a Signature Service that requires a sign message to be displayed as part of the authentication process will request a different authentication context URI (see section 7) than a typical Service Provider just requesting authentication of a user, even if the requested Level of Assurance is the same.

Identity Providers SHALL exclusively use one of the requested authentication contexts in `<saml2p:AuthnRequest>` in the `<saml2:AuthnContextClassRef>` element under the `<saml2p:RequestedAuthnContext>` element, when present, to determine the requested authentication process and Level of Assurance. The Identity Provider SHALL respond with an error `<saml2p:StatusCode>` with the value `urn:oasis:names:tc:SAML:2.0:status:Requester` [SAML2Core] if no requested authentication context is supported. If no requested authentication context is present in the `<saml2p:AuthnRequest>`, the Identity Provider MAY return the result of a default authentication process that is consistent with the Identity Providers metadata.

Note: The Identity Provider does not have to consider the service entity categories ([EidEntCat]) declared in the Service Provider's metadata entry when determining the requested authentication context under which the authentication should be performed. The purpose of the service entity categories is primarily to support service matching in discovery services and attribute release policies in Identity Providers. Significant Identity Provider products and software are not equipped to use service entity category information to determine the requested authentication context.

5.4.5 Single Sign On Processing

An Identity Provider conformant to this profile MAY issue an assertion relying on a previously established security context (active session) instead of authenticating the user. However, the Identity Provider MUST NOT re-use an already existing security context in the following cases:

- When the security context has expired, i.e., the time elapsed since the security context was established is too long given the SSO-policy stipulated by the federation.
- When the `<saml2p:AuthnRequest>` contains a `ForceAuthn` attribute with the value of `"true"`.
- If the original authentication process, which led to the establishment of the security context, was performed using a weaker Level of Assurance that what is requested in the current `<saml2p:AuthnRequest>` message.

If the Identity Provider user interface contains some sort of user consent, or information, concerning which attributes, or any other information, that is included in an assertion being issued, the Identity Provider SHOULD preserve this functionality if a `<saml2p:AuthnRequest>` message requesting a different set of attributes (or any other information) compared to what was delivered in the assertion at the time of establishing the security context. The Identity Provider may require re-authentication or display a user interface for consent/information in these cases.

6 Authentication Responses

6.1 Security Requirements

The endpoint(s) at which a Service Provider receives a `<saml2p:Response>` message MUST be protected by TLS/SSL ([SAML2Int] states SHOULD).

The `<saml2p:Response>` message issued by the Identity Provider MUST be signed using a `<ds:Signature>` element within the `<saml2p:Response>` element.

The `<saml2:Assertion>` element issued by the Identity Provider MAY be signed using a `<ds:Signature>` element within the `<saml2:Assertion>`. If a Service Provider requires signed assertions, by assigning the `WantAssertionsSigned` attribute of its metadata record (see chapter 2.1.2), the Identity Provider MUST sign assertions issued to this Service Provider (as well as the response message as stated above).

Identity Providers SHALL utilize XML Encryption and return a `<saml2:EncryptedAssertion>` element in the `<saml2p:Response>` message. The elements `<saml2:EncryptedID>` and `<saml2:EncryptedAttribute>` MUST NOT be used; instead the entire assertion should be encrypted.

Service Providers SHOULD NOT accept unsolicited `<saml2p:Response>` messages (i.e., responses that are not the result of an earlier `<saml2p:AuthnRequest>` message). Service Providers that do accept unsolicited response messages MUST ensure, by other means, that the security and processing requirements of this profile (section 6.3) can be fully satisfied. [SAML2Int] allows the use of unsolicited responses, but this profile has more strict security and processing requirements that make the use of unsolicited responses violate these requirements.

6.2 Message Content

The `<saml2:Response>` message MUST contain an `<saml2:Issuer>` element containing the unique identifier (entityID) of the issuing Identity Provider.

The `AuthnInstant` attribute of the `<saml2:AuthnStatement>` element MUST be assigned the time when the actual authentication took place. This time may differ from the `IssueInstant` attribute of the assertion itself, which holds the time when the assertion was issued. This is especially important in cases of re-use of already established security contexts at the Identity Provider side (Single Sign On).

Each identity assertion MUST have a `<saml:Subject>` element that specifies the principal that is the subject of all of the statements in the assertion.

The value of the `<saml:NameID>` element under the `<saml:Subject>` element MUST hold a pseudonym identifier of the subject, which SHALL be:

- Unique for the IdP – SP combination being the issuer and recipient for the assertion.
- Constructed in a manner that does not reveal the registered identity of the subject.

The `<saml2:Subject>` element MUST contain one `<saml2:SubjectConfirmation>` element containing a `Method` of `urn:oasis:names:tc:SAML:2.0:cm:bearer`. This element MUST contain a `<saml2:SubjectConfirmationData>` element that contains at least the following:

- An `InResponseTo` attribute matching the request's ID.
- A `Recipient` attribute containing the Service Provider's assertion consumer service URL (see sections 5.3 and 5.4.1).

- A `NotOnOrAfter` attribute containing a time instant at which the subject no longer can be confirmed.

The `<saml2:SubjectConfirmationData>` MUST also contain an `Address` attribute containing the network address from which an attesting entity (user) can present the assertion.

The assertion MUST contain a `<saml2:Conditions>` element containing the following attributes and elements:

- A `<saml2:AudienceRestriction>` element including the requesting Service Provider's unique identifier (entityID) as an `<saml2:Audience>` value.
- A `NotBefore` attribute specifying the earliest time instant at which the assertion is valid.
- A `NotOnOrAfter` attribute specifying the time instant when the assertion expires.

An Identity Provider conformant to this profile MUST, in its issued assertions, include an authentication context URI indicating under which Level of Assurance the assertion was issued. This identifier MUST be placed under the `<saml2:AuthnStatement>` element as the value of an `<saml2:AuthnContextClassRef>` element that is part of the `<saml2:AuthnContext>` element.

```
...
<saml2:AuthnStatement AuthnInstant="2013-03-15T09:22:00" SessionIndex="b07b804c-7c29-ea16-7300-4f3d6f7928ac">
  <saml2:AuthnContext>
    <saml2:AuthnContextClassRef>http://id.elegnamnden.se/loa/1.0/loa3</saml2:AuthnContextClassRef>
    ...
  </saml2:AuthnContext>
</saml2:AuthnStatement>
...
```

Example of how an Authentication Context URI identifier representing a Level of Assurance is included in an authentication statement.

6.2.1 Attribute Release Rules

An Identity Provider determines which attributes to include in the `<saml2:AttributeStatement>` element of an assertion based on the Service Provider requirements and its agreements with the user being authenticated. Service Provider attribute preferences and requirements are specified by the service entity categories [EidEntCat] and requested attributes in the `<md:AttributeConsumingService>` element declared in the Service Provider metadata. A service entity category specifies the attribute set (as defined in [EidAttributes]) that is requested for the attribute release process.

An Identity Provider declares service entity categories in order to publish its ability to deliver attributes according to certain attribute sets. For all declared service entity categories, the Identity Provider MUST possess the ability to deliver the mandatory attributes of the underlying attribute set. See [EidEntCat] and [EidAttributes] for details.

The Service Provider is responsible for checking that an Identity Provider is capable of providing necessary attributes before sending a request and to verify that it received all attributes necessary for providing a requested service. Checks whether an Identity Provider is capable of fulfilling the needs of a Service Provider can be done either by relying on a Discovery Service to filter out non-conformant Identity Providers, and/or by examining the metadata of Identity providers. An Identity Provider receiving a request for more attributes than it can provide SHOULD return an assertion with the attributes it can provide according to its defined attribute release policy, leaving it up to the Service Provider to decide how to proceed, e.g., by denying service to the authenticated user, provide limited services or to use other resources to collect necessary attributes.

6.3 Processing Requirements

This profile mandates a correct processing of a `<saml2p:Response>` message in order to ensure proper protection from the security threats described in [SAML2Sec]. Processing requirements are listed in [SAML2Core], [SAML2Prof] and [SAML2Sec]. This document will list the necessary requirements that apply to this profile.

After the Service Provider has encrypted the assertion from the received response message the following requirements apply. Any verification that fails MUST lead to that the Service Provider rejects the response message and does not use the assertion.

Some of the processing requirements below are defined in order to protect from MITM- or MITB-attacks[3] were unsigned authentication requests may be changed before being sent to the Identity Provider. However, a Service Provider MUST implement all of the specified processing requirements even if it sends signed authentication request messages.

6.3.1 Signature Validation

The signature present on the `<saml2p:Response>` message, and optionally on the `<saml2:Assertion>`, MUST be successfully verified.

The public key being used to verify the signature MUST appear in the issuing Identity Provider's metadata record (as a `<ds:X509Certificate>` or `<ds:KeyValue>` element under the `<ds:KeyInfo>` element).

6.3.2 Subject Confirmation

Based on the `InResponseTo` attribute of the `<saml2:SubjectConfirmationData>` the Subject Provider MUST be able to obtain the corresponding `<saml2p:AuthnRequest>` message, or a secure context containing corresponding information from the request (for future processing of the assertion).

The `Recipient` attribute from the bearer `<saml2:SubjectConfirmationData>` element MUST match the location to which the `<saml2p:Response>` message was delivered **and** match the value the `AssertionConsumerServiceURL` attribute included in the request message, or if this attribute was not provided in the request message, the default response location specified in the Service Provider's metadata entry, as described in section 5.4.2.

The time from the `NotOnOrAfter` attribute from the bearer `<saml2:SubjectConfirmationData>` MUST NOT have passed compared with the time instant at which the subject is confirmed (i.e., when the assertion is validated). A reasonable allowable clock skew between the providers should be taken in account.

If the `Address` attribute is assigned to the bearer `<saml2:SubjectConfirmationData>` element, the Service Provider MAY choose to check the user agent's client address against it. Practical issues regarding the Service Provider's network setup and the risk of introducing false negatives makes this an optional step in the validation phase.

6.3.3 Conditions

The Service Provider MUST assert that the value of the `<saml2:Audience>` element under the `<saml2:AudienceRestriction>` element matches the unique entityID of the Service Provider.

[3] MITM stands for "man in the middle" and MITB stands for "man in the browser".

The Service Provider MUST verify that the time instant at which the assertion is validated is within the range given by the `NotBefore` and `NotOnOrAfter` attributes of the `<saml2:Conditions>` element (allowing for a reasonable clock skew). See also the processing of the `NotOnOrAfter` attribute in section 6.3.2.

6.3.4 The Authentication Statement

The Service Provider MUST assert that the `<saml2:AuthnStatement>` contains a `<saml2:AuthnContext>` element that holds a `<saml2:AuthnContextClassRef>` element having as its value the authentication context URI indicating under which Level of Assurance the authentication was performed. The Level of Assurance declared in the assertion MUST be equal to, or stronger[4] than, the Level of Assurance requested by the Service Provider.

6.3.5 General Security Validation

In order to protect itself from replay attacks, the Service Provider MUST ensure that the same assertion is not processed more than once within the time it is valid (with respect to the `NotOnOrAfter` attribute of the `<saml2:Conditions>` element).

In order to prevent stolen assertions and user impersonation, the Service Provider SHOULD implement a validation that rejects an assertion if the time given it its `IssueInstant` attribute compared to the time when the response message is received is too great. This time is typically on the order of seconds, and limits the time window when a stolen assertion could be used.

If the Service Provider included the attribute `ForceAuthn` with a value of `"true"` in the authentication request, the Service Provider SHOULD ensure that the `AuthnInstant` attribute of the `<saml2:AuthnStatement>` element is greater than the time when the request was sent (allowing for a reasonable clock skew).

6.4 Error Responses

If the Identity Provider returns an error, it MUST NOT include any assertions in the `<saml2p:Response>` message.

An Identity Provider conformant with this profile SHOULD NOT make use of any other `<saml2p:StatusCode>` values than those specified in section 3.2.2.2 of [SAML2Core], and the top-level `<saml2p:StatusCode>` value may only be one of the following error identifiers:

- `urn:oasis:names:tc:SAML:2.0:status:Requester` – The request could not be performed due to an error on the part of the Service Provider.
- `urn:oasis:names:tc:SAML:2.0:status:Responder` – The request could not be performed due to an error on the part of the Identity Provider.
- `urn:oasis:names:tc:SAML:2.0:status:VersionMismatch` – The Identity Provider could not process the request because the version of the request message was incorrect.

If an Identity Provider displays information describing an error in its user interface it MUST also offer ways for the end user to confirm this information (for example, by including an OK-button). When the end user confirms taking part of the information (i.e., clicks on the OK-button), the `<saml2p:Response>` message is posted back to the Service Provider according to the HTTP POST binding [SAML2Bind].

[4] A stronger Level of Assurance identifier is simply a LoA having a higher value than what it is compared with, i.e., http://id.elegnamnden.se/loa/1.0/loa4 is stronger than http://id.elegnamnden.se/loa/1.0/loa3.

7 Authentication for Signature

"DSS Extension for Federated Central Signing Services", [EidDSS], defines an extension to the OASIS DSS protocol for providing centralized Signature Services within the Swedish eID Framework. This specification defines the communication between a *Signature Requestor*[5] and a Signature Service, but does not cover SAML specific requirements regarding the user authentication phase that is part of the signature process.

This section defines requirements on the SAML authentication process when authentication is requested by a Signature Service, acting as a SAML Service Provider. All requirements regarding user authentication specified earlier in this profile are still valid. This section extends these requirements for the "authentication for signature" process.

7.1 Authentication Context URIs for Signature Services

The Swedish eID Framework defines additional authentication context URIs to be used in `<saml2p:AuthnRequest>` and `<saml2:Assertion>` elements during "authentication for signature". These authentication context URIs are applicable when the Identity Provider is required to display a sign message as part of the authentication process. These URIs are:

- http://id.elegnamnden.se/loa/1.0/loa2-sigmessage
- http://id.elegnamnden.se/loa/1.0/loa3-sigmessage
- http://id.elegnamnden.se/loa/1.0/loa4-sigmessage
- http://id.elegnamnden.se/loa/1.0/eidas-low-sigm
- http://id.elegnamnden.se/loa/1.0/eidas-sub-sigm
- http://id.elegnamnden.se/loa/1.0/eidas-high-sigm
- http://id.elegnamnden.se/loa/1.0/eidas-nf-sub-sigm
- http://id.elegnamnden.se/loa/1.0/eidas-nf-high-sigm

These URIs extend the corresponding authentication context URIs used to represent Level of Assurance identifiers (see section 3.1.1 of [EidRegistry]) with requirements listed in the sections below. A Signature Service MAY use any of the defined authentication context URIs. The URIs listed above are only used when there is an explicit requirement for the Identity Provider to display a sign message provided in the authentication request.

7.2 Authentication Requests

Authentication requests from a Signature Service SHALL meet the following requirements:

- The `ForceAuthn` attribute of the `<saml2p:AuthnRequest>` element MUST be set to `"true"`.
- The `<saml2p:AuthnRequest>` element MUST be signed. This MUST also be indicated in the Signature Service metadata record using the `AuthnRequestsSigned` attribute (see section 2.1.4).

An Identity Provider that accepts an `<saml2p:AuthnRequest>` message from a Service Provider that has indicated that it is a Signature Service[6] MUST provide a user interface that is indicating that the end user is performing a signature.

[5] A *Signature Requestor* is a Service Provider within the federation to which the user previously has logged in to and from where the user initiates a signature operation.

[6] An Identity Provider identifies a Service Provider as a Signature Service if it declares the http://id.elegnamnden.se/st/1.0/sigservice URI as a service type entity category in its metadata (see 2.1.4).

7.2.1 Requesting Display of Signature Message

[EidDSS_Profile] specifies that a Signature Requestor may include a `SignMessage` element (as defined by [EidDSS]) in a signature request. This element holds a message that the Identity Provider, which is responsible for "authentication for signature", should present to the user that is performing the signature.

A Signature Service MAY request the Identity Provider to show a sign message to the user by including the `SignMessage` element from the signature request as a child element to an `<saml2p:Extensions>` element in the `<saml2p:AuthnRequest>` message (see section 3.2.1 of [SAML2Core]).

If the `SignMessage` element from the signature request includes a `MustShow` attribute with the value `true`, the Signature Service MUST require that the provided sign message is displayed by the Identity Provider, by including an authentication context URI (as defined in section 7.1 above) to the `<saml2:AuthnContextClassRef>` element that is part of the `<saml2p:RequestedAuthnContext>` element of the `<saml2p:AuthnRequest>` message.

Identity Providers SHALL advertise supported authentication contexts defined by the URIs listed in section 7.1, by including the URIs of supported authentication contexts as `EntityAttributes` of the type `urn:oasis:names:tc:SAML:attribute:assurance-certification` in its metadata.

```
...
<md:Extensions>
  <mdattr:EntityAttributes xmlns:mdattr="urn:oasis:names:tc:SAML:metadata:attribute">
    <saml:Attribute Name="urn:oasis:names:tc:SAML:attribute:assurance-certification"
                    NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
      <saml:AttributeValue type="xs:string">http://id.elegnamnden.se/loa/1.0/loa3</saml:AttributeValue>
      <saml:AttributeValue type="xs:string">http://id.elegnamnden.se/loa/1.0/loa3-sigmessage</saml:AttributeValue>
    </saml:Attribute>
    ...
  </mdattr:EntityAttributes>
</md:Extensions>
...
```

Example of how an Identity Provider advertises its support for LoA3 authentication (including support for displaying of sign messages).

Identity Providers processing a request with a requested authentication context identified by any of the URIs listed in 7.1 SHALL meet the following requirements (in addition to other general requirements associated with requests from signature services:

- The authentication request SHALL contain a sign message that can be extracted by the Identity Provider. If the Identity Provider fails to locate, decrypt or extract the sign message in clear text form, it must return an error response.
- The Identity Provider MUST display the sign message to the user in a manner that is consistent with the data format of the sign message. If necessary, the Identity Provider MUST process defined filtering rules on the message. If the present message format is not supported or the sign message for any reason cannot be displayed in a proper manner, the Identity Provider must return an error response.
- If authentication and sign message confirmation by the user was successful, the Identity Provider MUST include the authentication context URI from the list in 7.1 in the assertion that is consistent with the authentication context requested in the authentication request.
- The Identity Provider MUST NOT return an assertion without performing authentication process consistent with the requested authentication context which includes display of a sign message, even if the request has no present `ForceAuthn` attribute or includes a `ForceAuthn` attribute set to the value `"false"`.

7.3 Authentication Responses

By including an authentication context URI listed in section 7.1 (sign message URI) in SAML assertion under the `<saml2:AuthnContextClassRef>` element of the `<saml2:AuthnStatement>` element in the response, the Identity Provider asserts that it has successfully displayed the sign message received in the request for the user and that the user has accepted to sign under the context of this sign message[7].

An Identity Provider MUST NOT return an authentication context URI in an assertion, other than those listed in section 7.1, if the request included one of these URIs as the requested authentication context. If the Identity Provider failed to display the sign message or the user failed to accept it, and the request indicated that the sign message MUST be displayed, then the Identity Provider MUST return an error response with the status code `urn:oasis:names:tc:SAML:2.0:status:AuthnFailed`.

[7] As defined in section 5.3, only exact matching of authentication context URIs are allowed. As a consequence the Identity Provider can only assert a sign message authentication context URI according to section 7.1 if such an authentication context was requested in the authentication request. It is therefore the responsibility of the Signature Service requesting authentication to always request a sign message authentication context if it requires evidence that the sign message has been displayed to the user.

8 Normative References

[RFC2119]
Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, March 1997.

[SAML2Int]
SAML2int profile v0.21 – SAML 2.0 Interoperability Profile.

[SAML2Core]
OASIS Standard, Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005.

[SAML v2.0 Errata 05]
SAML Version 2.0 Errata 05. 01 May 2012. OASIS Approved Errata.

[SAML2Bind]
OASIS Standard, Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005.

[SAML2Prof]
OASIS Standard, Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005.

[SAML2Meta]
OASIS Standard, Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005.

[SAML2Sec]
Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005.

[SAML2IAP]
SAML V2.0 Identity Assurance Profiles Version 1.0, 05 November 2010.

[MetaIOP]
OASIS Committee Specification, SAML V2.0 Metadata Interoperability Profile Version 1.0, August 2009.

[SAML2MetaUI]
OASIS Draft, SAML V2.0 Metadata Extensions for Login and Discovery User Interface Version 1.0, September 2010.

[SAML2MetaAttr]
OASIS Committee Specification, SAML V2.0 Metadata Extension for Entity Attributes Version 1.0, August 2009.

[EntCat]
The Entity Category SAML Entity Metadata Attribute Type, March 2012.

[IdpDisco]
OASIS Committee Specification, Identity Provider Discovery Service Protocol and Profile, March 2008.

[EidRegistry]
Registry for identifiers assigned by the Swedish e-identification board.

[EidAttributes]
Attribute Specification for the Swedish eID Framework.

[EidTillit]
Tillitsramverk för Svensk E-legitimation.

[EidEntCat]
Entity Categories for the Swedish eID Framework.

[EidDisco]
Discovery within the Swedish eID Framework.

[EidDSS]
DSS Extension for Federated Central Signing Services - Version 1.1.

[EidDSS_Profile]
Implementation Profile for using OASIS DSS in Central Signing Services.

9 Changes between versions

Changes between version 1.3 and version 1.4:

- Version 1.3 of this profile stated that a `<saml2p:AuthnRequest>` message MUST contain an `AssertionConsumerServiceURL` attribute identifying the desired response location. It has shown that this requirement aggravates interoperability since some of the major providers of Service Provider software do not fully support this attribute. Furthermore, the requirement does increase security since an Identity Provider may only post response messages to locations registered in the `<md:AssertionConsumerService>` elements of the Service Provider metadata entry. Therefore, chapter 5.3, "Message Content", has been changed to state that the `<saml2p:AuthnRequest>` message SHOULD contain an `AssertionConsumerServiceURL` attribute. Changes have also been made to sections 5.4.2 and 6.3.2 where processing requirements were updated.
- In section 5.3, a clarification regarding specifying more than one authentication context URI was made.
- In section 7.1, a set of authentication context URIs for the eIDAS Framework was added.

Changes between version 1.2 and version 1.3:

- This profile now extends a newer version of the SAML2Int Deployment Profile (see http://saml2int.org/profile/current/).
- Clarifications on how entity categories are represented in metadata were made to chapters: 2.1.2, 2.1.3, and 2.1.4.
- Changes were made to chapter 6.1, "Security Requirements", where the profile now requires the entire `<saml2p:Response>` message to be signed, as compared to the previous version where the signature requirement was put on `<saml2:Assertion>` elements.
- In chapter 6.2, it is now specified that an `Address` attribute MUST be part of the `<saml2:SubjectConfirmationData>` element. The previous version stated SHOULD.
- Chapter 6.2.1, "Attribute Release Rules", was introduced to clarify how the attribute release process should be handled by an issuing entity.
- A Service Provider is now obliged to explicitly specify the required Level of Assurance under which a specific authentication should be performed. This is specified in chapter 5.3, "Message Content", and 5.4.4, "Authentication Context and Level of Assurance Handling".
- The specification "Authentication Context Classes for Levels of Assurance for the Swedish eID Framework" has been removed from the Swedish eID Framework. The reason for this is that it was proven difficult to make use of the `<saml2:AuthnContextDecl>` element to store authentication context parameters, and that no commercial, or open source, Identity Provider software had support for this feature. [EidAttributes] now describe how the `authContextParams` attribute may be used for the same purpose, and the examples where this information was stored under the `<saml2:AuthnContextDecl>` element was removed from chapter 6.2, "Message Content".
- Chapter 7, "Authentication for Signature", was introduced to specify requirements regarding the process of "authentication for signature" where a *Signature Service* requests that a user performing a signature authenticates.

Changes between version 1.1 and version 1.2:

- This profile now explicitly defines requirements for the use of signed authentication request messages, see sections 2.1and 5.2.
- This profile now allows the HTTP-POST binding to be used for sending authentication request messages (see chapter 5.2, "Binding and Security Requirements"). The main reason for this is to facilitate the use of signed authentication request messages.
- In chapter 5.4, additional processing requirements for received authentication requests were added or changed. These include:

  - Validation of assertion consumer addresses (5.4.1).
  - Clarifications to chapter 5.4.4.
  - Single Sign On processing (5.4.5).
- This profile now states that “Unsolicited response” messages are not accepted by Service Providers due to security reasons, see chapter 6.1, “Security Requirements”.
- Changes and additions in chapter 6.2, “Message Content”, for responses including:
  - Clarifications about the usage of the `AuthnInstant` attribute of the `<saml2:AuthnStatement>` element.
  - Specifications of the use of `<saml2:SubjectConfirmation>` in assertions.
  - Clarifications on the use of audience restrictions and assertion validity.
- Chapter 6.3, “Processing Requirements”, was added. This chapter contains specifications and requirements of how a response message should be processed in order to maintain security.

Changes between version 1.0 and version 1.1:

- In chapter 5.1, “Discovery”, a reference to the specification “Discovery within the Swedish eID Framework” [Eid2Disco] was added.
- In chapter 5.4.4, a note was added that informs about the need to ensure IdP-capabilities regarding level of assurance before issuing a request.
- In chapter 6.2, “Message Content”, an example of how an Identity Provider may include an authentication context class declaration was provided.
- Some faulty references were corrected.