Implementation Profile for BankID Identity Providers within the Swedish eID Framework

Version 1.2 - 2019-11-01 - Draft version

ELN-0612-v1.2

Table of Contents

1. Introduction

This profile defines how a SAML Identity Provider that offers authentication using the Swedish BankID technology should implement its services to be compliant with the Swedish eID Framework. It extends the "Deployment Profile for the Swedish eID Framework", [EidProfile], with requirements and recommendations for Identity Providers offering BankID authentication and signature services.

The BankID interface for authentication and signature, the Relying Party Interface, is described in the "BankID Relying Party Guidelines", [BankID_Spec], specification. This specification MUST be fully implemented and supported by BankID Identity Providers compliant with the Swedish eID Framework specifications.

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:Element>` – for elements from the SAML 2.0 Protocol namespace.
- `<saml2:Element>` – 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:Element>` – for elements defined in [SAML2Meta].
- `<mdattr:Element>` – for elements defined in [SAML2MetaAttr].

1.3. BankID Methods and Applications

There are three types of BankID:

- Mobile BankID - End users use the "BankID app" on their mobile devices to authenticate or perform a signature. In these cases the user certificate is stored in the app and protected by a personal code.
- BankID on file - End users use the desktop program "BankID Security Application" to authenticate or perform a signature. The user certificate is stored in a file on the computer and is protected by a user password.
- BankID on card - End users make use of the same desktop program as described above, but the certificate is placed on a smart card. The user private key is unlocked using the PIN-pad on the smart card reader.

The three above methods are all "BankID", but historically, relying parties have made a difference between "Mobile BankID" and "BankID" (the original desktop version).

1.3.1. Representation as Identity Providers

An actor offering BankID services can choose to use **one** BankID Identity Provider supporting all different BankID methods, or use **several** Identity Provider instances, one for each BankID method.

Services that support all methods within one Identity Provider instance usually displays a question to the user before authentication starts, where the user chooses between "Using BankID on this device or another device". In an environment where a discovery service (or similar) is being used, this means that the user has to make two choices before the actual authentication process starts; first at the discovery service where the user selects "BankID" and then at the BankID Identity Provider where the user selects the type of BankID authentication to use.

It is RECOMMENDED that BankID services are split into separate Identity Providers for each supported BankID method. The reasons for this are the above argument about discovery, but also the fact that a Service Provider should be able to select which type of authentication that is required (for example, Mobile BankID may be accepted but not BankID on file).

1.3.2. Recommended Limitations

The table below states the RECOMMENDED support and behaviour when support for BankID is implemented using separate Identity Providers (as recommended in section 1.3.1 above).

Identity Provider	**Desktop**	**Mobile Phone**	**Tablet**
Mobile BankID	Start BankID on other device[1] (mobile phone or tablet).	Start BankID on the same device[2].	Prompt the user to ask whether to start BankID on the tablet or on another device[3] (mobile phone).
BankID on file (*or on card*)	Start BankID on the same device[4].	Not supported[5].	Not supported[5].

1. The user initiates a BankID operation from his or hers desktop computer and selects to use Mobile BankID. In this case the Mobile BankID app is started on another device (since Mobile BankID does not exist on desktop computers).
2. The user initiates a BankID operation from his or hers mobile phone and selects to use Mobile BankID. In this case the BankID app is started on the same device. It is highly unlikely that a user uses one mobile phone to visit a service and wants to use his or hers BankID on another device.
3. The user initiates a BankID operation from his or hers tablet and selects to use Mobile BankID. In this case the recommendation is to prompt the user to ask whether the Mobile BankID app should be automatically started on the tablet, or if the user wishes to use BankID on another device (probably a mobile phone). The reason for this recommendation is that most users have a BankID on their mobile phones, but not necessarily on their tablets.
4. The user initiates a BankID operation from his or hers desktop computer and selects to use BankID on file. The BankID Security Application is started on the same computer. It is not a likely use case to use one computer to connect to the service and another one for BankID.
5. This case should not be supported. If the user selects "BankID on file" from a mobile phone or tablet, the Identity Provider should display an error message stating that Mobile BankID should be used instead and post an error response back to the Service Provider.

Note: Items 4 and 5 above also apply to BankID on card. A service MAY choose to implement BankID on file and BankID on card as separate Identity Providers or as one Identity Provider instance.

For Identity Providers implementing BankID support in **one** Identity Provider instance it is RECOMMENDED to make the assumption that the BankID app should be started on the same device if the user connects via a mobile phone.

1.4. Relying Party Configuration

When a Relying Party uses the BankID Relying Party API directly in order to implement BankID services, it has a set of configuration settings to choose from (see section 14.5 of [BankID_Spec]). Examples are:

- Should fingerprints be allowed instead of the user entering his or hers security code?
- Active certificate policies.
- Smart card reader preferences.

However, the services of a BankID Identity Provider are used by several parties within the federation and it is thus harder to maintain a per Service Provider configuration. Therefore, a BankID Identity Provider compliant with this profile SHOULD use the same configuration defaults as stated in section 14.5 of [BankID_Spec]).

> **Note:** It is of course allowed for a BankID Identity Provider to maintain specific Service Provider configurations, but this is outside of the scope for this profile, and there will be no specification support to accomplish this.

Besides the above Relying Party configurations, the BankID API offers two different ways to initiate a BankID operation for the cases where the user agent and the BankID app is not on the same device.

- A QR code is displayed in the UI and the user is prompted to scan this code using his or hers mobile device (see section 4 of [BankID_Spec]).
- The user is prompted for his or hers personal number.

The QR code functionality is a relatively new feature that was introduced to provide protection from fraudsters that remotely attempts to persuade people to authenticate or sign using their mobile BankID:s. By prompting the user to scan a QR code the user agent and mobile device are tied to the same physical location, and these kinds of frauds are eliminated.

Due to large number of BankID users and the fear of changing a well established pattern, many BankID Relying Parties have not yet implemented the use of QR codes. It is a consideration of ease of use versus higher security, and different service providers may have different opinion regarding the feature.

A Service Provider making use of a BankID Identity Provider has the possibility to explicitly request that QR codes, instead of user ID prompting, is used by the Identity Provider to initiate BankID operations. This is performed by declaring the `http://id.swedenconnect.se/general-ec/1.0/secure-authenticator-binding` entity category (see section 6.1 of [EidEntCat]) in the Service Provider metadata.

A BankID Identity Provider compliant with this profile SHOULD support the use of QR codes to initialize BankID operations.

BankID Identity Providers that support the QR code feature MUST declare the `http://id.swedenconnect.se/general-ec/1.0/secure-authenticator-binding` entity category in its metadata and MUST check the presence of this entity category from a requesting Service Provider's metadata when processing a request, and MUST prompt the user to scan a QR code instead of asking for the personal identity number if the entity category is present.

> See section 6, "Metadata", for details how to declare the secure-authenticator-binding entity category.

2. Attributes

An BankID Identity Provider use the BankID Relying Party API, as described in [BankID_Spec], to communicate with the BankID-server when providing its services to end users. When a BankID-operation has completed successfully, the Identity Provider (the BankID Relying Party) invokes the Collect-method (`/rp/v5/collect`) to obtain the result from the operation.

The table below contains attribute transformation mappings between attributes from a Collect-method response as described in section 14.2.5 of [BankID_Spec] and attributes defined within the Swedish eID Framework as defined in [EidAttributes].

An Identity Provider should not necessarily release all transformed attributes received from the BankID-server to the Service Provider. See further section 5.1, "Attribute Release Rules".

2.1. Attribute Transformation

BankID attribute	SAML Attribute	Description
`orderRef`	transactionIdentifier `urn:oid:1.2.752.201.3.2`	The BankID order reference received from a BankID Auth (`/rp/v5/auth`) or Sign (`rp/v5/sign`) method invocation. This parameter is supplied as an input parameter to the Collect-call and is the unique transaction identifier for the BankID-operation.
`completionData.user.personalNumber`	personalIdentityNumber `urn:oid:1.2.752.29.4.13`	Swedish ”personnummer”. 12 digits without hyphen.
`completionData.user.givenName`	givenName `urn:oid:2.5.4.42`	User's given name.
`completionData.user.surname`	sn `urn:oid:2.5.4.4`	User's surname.
`completionData.user.name`	displayName `urn:oid:2.16.840.1.113730.3.1.241`	User's given name and surname.
`completionData.cert.notBefore`	*bankidNotBefore* key in authContextParams `urn:oid:1.2.752.201.3.3`	Start of validity of user's BankID. No direct attribute mapping exists, but may be represented as key-value pair in authContextParams, where the key is *bankidNotBefore*, see 2.1.1 below.
`completionData.cert.notAfter`	*bankidNotAfter* key in authContextParams `urn:oid:1.2.752.201.3.3`	End of validity of user's BankID. No direct attribute mapping exists, but may be represented as key-value pair in authContextParams, where the key is *bankidNotAfter*, see 2.1.1 below.
`completionData.device.ipAddress`	*bankidUserAgentAddress* key in authContextParams `urn:oid:1.2.752.201.3.3`	The IP-address of the user agent presented to the BankID server. In cases where a user uses BankID "on another device" this address may not be the same as the web user agent. No direct attribute mapping exists, but may be represented as key-value pair in authContextParams, where the key is *bankidUserAgentAddress*, see 2.1.1 below.

BankID attribute	SAML Attribute	Description
`completionData.signature`	userSignature `urn:oid:1.2.752.201.3.11`	The signature applied by the user as part of the authentication/signature process.
`completionData.ocspResponse`	authServerSignature `urn:oid:1.2.752.201.3.13`	The OCSP response signed by the BankID issuer that proves that the user BankID was checked for revocation.

2.1.1. The authContextParams Attribute

The `authContextParams` attribute, see section 3.2.1 of [EidAttributes], is a general purpose attribute to be used when non-standardized authentication data is to be transfered in a SAML assertion.

The attribute is used by attribute providers to release data from an authentication process that has no attribute definition of its own. Thus, should the BankID attributes `completionData.cert.notBefore`, `completionData.cert.notAfter` and `completionData.device.ipAddress` be transformed and included into an assertion, they would have to be placed as key-value pairs of the `authContextParams` attribute as the example below.

```
<saml2:Attribute xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                         FriendlyName="authContextParams"
                         Name="urn:oid:1.2.752.201.3.3"
                         NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
  <saml2:AttributeValue xsi:type="xs:string">
    bankidNotBefore=2016-05-30T09%3A30%3A10Z;bankidNotAfter=2018-05-30T09%3A30%3A10Z;bankidUserAgentAddress=85.229.202.232
  </saml2:AttributeValue>
</saml2:Attribute>
```

The example above represents the following BankID attributes and values:

- `completionData.cert.notBefore` = 2016-05-30T09-30-10Z
- `completionData.cert.notAfter` = 2018-05-30T09-30-10Z
- `completionData.device.ipAddress` = 85.229.202.232

> The format for the notBefore and notAfter attributes should be the representation as given by the XML type `xs:dateTime`.

3. Identity Provider User Interface

This profile does not state any requirements on how the user interface for an Identity Provider implementing BankID services should be implemented other than the statements listed in the sub sections below.

3.1. General Requirements

The user interface for a BankID Identity Provider SHOULD use the recommended user and error messages as defined in sections 6, "Recommended User Messages", and 11, "Recommended Terminology", of [BankID_Spec].

The user interface for a BankID Identity Provider MUST display information about the Service Provider that sent the request. It is RECOMMENDED that this information is obtained from the `<mdui:UIInfo>` element from the Service Provider's metadata entry.

It MUST be clear to the user whether an authentication or a signature process is ongoing.

When an error occurs during an authentication or signature operation, the Identity Provider MUST display an error message that can be easily understood by the end user, and offer the possibility to acknowledge the error so that an error response may be posted back to the requesting Service Provider (as specified in section 6.4, "Error Responses", of [EidProfile]).

3.2. Automatic Start of the BankID Client

When an operation is initiated where the BankID client (app or desktop program) is on the same device as the user agent (web browser) the Identity Provider SHOULD attempt to "auto start" the BankID client as described in [BankID_Spec].

For the above cases where the BankID client is automatically started from the Identity Provider, the Identity Provider user interface SHOULD NOT ask for the user's personal identity number. This information is available in the "BankID app" or "BankID Security Application".

Auto starting the BankID app from a mobile device requires the built-in web browser to be used to guarantee full support, see section 3.1 of [BankID_Spec]. If the Identity Provider detects that the user is not using the platform's default browser it SHOULD prompt for the personal identity number and ask the user to manually start the BankID app (and switch back to the browser after the BankID operation).

> **Note:** As an alternative to the above requirement the Identity Provider may display a button with an autostart link that starts the BankID app when clicked. The personal identity number will not be needed and the user will not have to start the app manually.
>
> However, on iOS this will not work if an embedded browser is running within an app, since apps that are started by apps need to be whitelisted. Another thing to take into account is that the user still has to switch back from the BankID app to the browser, and that may not be obvious for the user (since the app was started automatically).
>
> The recommendation is to abstain from using autostart links/buttons.

3.3. Mobile BankID on another Device

If the user agent (web browser) and the BankID app is not on the same device, a BankID Identity Provider that supports the BankID QR code feature MUST check for the presence of the `http://id.swedenconnect.se/general-ec/1.0/secure-authenticator-binding` entity category in a requesting Service Provider's metadata entry. If the Service Provider has declared this entity category, the BankID Identity Provider MUST display a generated QR code for the user and MUST NOT prompt the user for the personal identity number.

> Note: If the `secure-authenticator-binding` entity category is present in the Service Provider metadata entry **and** a `<psc:PrincipalSelection>` extension containing a personal identity number is included in the authentication request from the Service Provider, the BankID Identity Provider SHOULD include this number in the BankID operation (`auth` or `sign`) along with the BankID requirement parameter `autoStartTokenRequired=true` when initializing the BankID operation (see chapter 4 in [BankID_Spec]). A generated QR code MUST still be displayed by the BankID Identity Provider, but by providing the personal identity number in the initializing BankID call the session will be bound not only to the BankID app that is used to scan the QR code but also to a specific user. If a user with a different personal identity number scans the QR code the operation will fail.

If the `secure-authenticator-binding` entity category is not declared by the requesting Service Provider, the Identity Provider needs to prompt the user for his or hers personal identity number (personnummer) in order to initiate a BankID operation.

Before prompting the user the Identity Provider SHOULD check if the current authentication request contains a `<psc:PrincipalSelection>` extension holding a personal identity number attribute value.

If the personal identity number is present in the `<psc:PrincipalSelection>` extension and the current operation is an "authentication for signature" operation (see section 4.2 below), the Identity Provider MUST NOT prompt the user for the personal identity number, but use the value received in the request to initiate the signature operation.

If the current operation is an "ordinary" authentication and the personal identity number is received in the request, the Identity Provider MAY use this value to pre-fill the personal identity number in the prompt dialogue to make the user experience smooth, but it SHOULD NOT use it to initiate the BankID operation without user interaction.

> Note: Until the `<psc:PrincipalSelection>` extension is widely used the Identity Provider MAY save the personal identity number in the user's web browser session storage (in a session cookie or more preferably using the HTML5 sessionStorage object). That way the Identity Provider can avoid prompting the user for the personal identity number for signature operations even if the requesting Signature Service does not support the `<psc:PrincipalSelection>` extension.

See also section 4.2.2, "Mobile BankID and the personNumber attribute".

3.3.1. User Experience Recommendations

In a scenario where the user first logs in to a service, and later performs a signature, care should be taken to the user experience versus security. The user will probably think is disturbing to have to scan a QR code for every signature he or she performs within the logged in session. If the service can protect against the remote fraudster threat by using QR code for login, and if the `<psc:PrincipalSelection>` extension preventing personal identity number prompting is used for subsequent signatures, we probably have found the safest and most user friendly process.

This could be accomplished by declaring the `secure-authenticator-binding` entity category for the Service Provider responsible of user login, but not declaring it for the service's Signature Service. Instead the Signature Service makes sure to always include the `<psc:PrincipalSelection>` extension in authentication requests sent.

Note: An Identity Provider processing a request from a signature service can derive the QR versus prompting for personal identity number setting for the corresponding "login service" if the `RequesterID` element is present in the authentication request. This element holds the entityID for the login service that initiated the signature operation.

3.4. Cancelling an Operation

A BankID Identity Provider MUST include a Cancel-button in the user interface enabling the possibility for the user to cancel the BankID operation.

If the use clicks the Cancel-button after a BankID-operation has been started[1] the Identity Provider MUST invoke the BankID-operation `/rp/v5/cancel`. Failure to do so may lead to a dangling BankID session that needs to time out before the user can use BankID again.

> [1]: Meaning that `/rp/v5/auth` or `/rp/v5/sign` has been called for the transaction.

4. Authentication Requests

4.1. Binding and Security Requirements

An Identity Provider conformant with this profile MUST require `<saml2p:AuthnRequest>` messages to be signed (by indicating this in its metadata, see section 6, "Metadata"). Thus, the Identity Provider 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.

4.2. Authentication for Signature

An Identity Provider conforming to the Swedish eID Framework is obliged to handle requests received from Signature Services as described in section 7, “Authentication for Signature”, of [EidProfile]. This section further specifies how a BankID Identity Provider should support “authentication for signature”.

A BankID Identity Provider that receives an `<saml2p:AuthnRequest>` message from a Signature Service MUST initiate a BankID **signature** operation. It MUST NOT initiate a BankID authentication operation for several reasons:

- the user interface in the BankID client (app or Desktop program) during authentication indicates that the user is logging on (and not signing which is the case when a request from a Signature Service is being processed),
- the user expects to be displayed a text describing what he or she is signing,
- and most importantly, BankID is PKI-based and has support for signing using a non-repudiation key, so there is no reason not to use this function.

The BankID client (app or desktop program) comprises a text box in which the signature message is displayed for the user. A BankID Identity Provider MUST NOT display the signature message in any other way than in this text box. How the signature message is assigned is specified below.

If the BankID Identity Provider prompts the user for his or hers personal identity number in order to initialize the BankID signature operation, the Identity Provider MUST honour the requirement that ensures that a sign message is only displayed to the intended principal, see section 7.2.1 in [EidProfile].

4.2.1. Input to BankID Signing

An Identity Provider that processes an `<saml2p:AuthnRequest>` from a Signature Service is not given the actual data that is being signed by the user via the Signature Service. However, in order to invoke the BankID signature function, the Identity Provider must supply the BankID-server with data to be signed. This section specifies the input to the BankID signature operation.

The "To-be-signed" data that is passed as input the the BankID Sign-method (`/rp/v5/sign`) is a combination of the data from the `userVisibleData` and `userNonVisibleData` parameters (section 14.1.2 of [BankID_Spec]).

4.2.1.1. userVisibleData - Signature Message

The Sign-method parameter `userVisibleData` holds data that will be signed by the user but it is also displayed in the BankID application text box.

If the `<saml2p:AuthnRequest>` message contains a `SignMessage` extension, the contents of this message MUST be assigned to the `userVisibleData` parameter (after necessary encoding).

A BankID Identity Provider MUST only process `SignMessage` elements having their `MimeType` attribute set to `text`[1] . For any other values (`text/html` or `text/markdown`), the Identity Provider MUST respond with an error.

If the `<saml2p:AuthnRequest>` message does not contain a `SignMessage` extension, the Identity Provider MUST assign a sensible default signature message to the `userVisibleData` parameter. How this message is constructed is the responsibility of the Identity Provider, but it must be obvious for the user who is the requesting party, i.e., the Service Provider that has ordered the signature operation[2].

> [1]: If the `MimeType` attribute is not set, `text` is the default value.
>
> [2]: For this purpose, the `<mdui:DisplayName>` element of the Signature Service's metadata entry, is a good and generic choice.

4.2.1.2. userNonVisibleData

In order to produce a BankID signature that contains a binding to the `<saml2p:AuthnRequest>` message that initiated this signature, a BankID Identity Provider compliant to this profile MUST assign the `userNonVisibleData` parameter with data that uniquely binds the signature to the `<saml2p:AuthnRequest>` message.

It is RECOMMENDED that the following function is used to produce this unique binding:

```
Base64Encode("entityID=" + URLEncode(<entityID of SP>) + ";" + "authnRequestID=" + URLEncode(<ID of AuthnRequest>))
```

4.2.2. Mobile BankID and the personNumber attribute

When Mobile BankID is being used to sign data and the user has initiated the signature operation against the Signature Service from another device (desktop och tablet), and the BankID QR code functionality is not being used, the `personNumber` parameter must be assigned in the BankID Sign-call.

Preferably this information was received in the `<psc:PrincipalSelection>` of the `<saml2p:AuthnRequest>` as described in section 3.3, "Mobile BankID on another Device".

Identity Providers wanting to support Signature Services that do not include the `<psc:PrincipalSelection>` extension MAY store the personal identity number in the user's web browser session storage during authentication, and use it during signature[1], see section 3.3 above.

> [1]: Almost all use cases where a user signs data is preceded by a login (authentication).

5. Authentication Responses

5.1. Attribute Release Rules

Section 2.1, "Attribute Transformation", specifies how BankID attributes should be transformed into SAML attributes defined in [EidAttributes]. However, it does not specify the attribute release rules stating which attributes that are to be released based on a particular request.

A BankID Identity Provider compliant to the Swedish eID Framework MUST honor the attribute release rules specified in section 6.2.1, "Attribute Release Rules", of [EidProfile]. This section further extends these rules with the following:

- A BankID Identity Provider SHOULD include the `transactionIdentifier`-attribute (a mapping of the BankID `orderRef`-attribute) in the `<saml2:AttributeStatement>` element independently of which attribute set that is requested. This attribute links the BankID operation to the assertion.
- It is RECOMMENDED that a BankID Identity Provider includes the `userSignature`-attribute (containing the BankID signature) in the `<saml2:AttributeStatement>` element when a BankID signature operation has been performed.
- Unless explicitly required[1] by the Service Provider the Identity Provider SHOULD NOT release any other attributes than those specified by the current attribute set(s)[2].

> [1]: A Service Provider explicitly requests attributes by declaring them as requested attributes in the `<md:AttributeConsumingService>` element of the Service Provider's metadata entry. See section 6.1.
>
> [2]: Based on the service entity categories that a Service Provider has declared in its metadata, an Identity Provider derives which attribute sets to apply during attribute release.

5.2. Error Responses

A BankID Identity Provider MUST map errors received from the underlying BankID-server into SAML error response messages where the top level status code is either:

- `urn:oasis:names:tc:SAML:2.0:status:Requester` - for errors that are due to authentication or signature failures or faults due to an error on the part of the Service Provider,
- `urn:oasis:names:tc:SAML:2.0:status:Responder` - for errors that are due to an internal, or technical, error in the BankID-server or Identity Provider.

Before a `<saml2p:Response>` message is posted back to the Service Provider the Identity Provider MUST display a relevant error message to the user.

It is RECOMMENDED that authentication/signature errors and failures to start the BankID client are represented using the second level status code `urn:oasis:names:tc:SAML:2.0:status:AuthnFailed`.

If the user cancels a BankID operation, either by clicking the Cancel-button in the Identity Provider user interface or the Cancel-button in the BankID app/Security Application, the Identity Provider SHOULD respond with a `<saml2p:Response>` message where the second level status code is `http://id.elegnamnden.se/status/1.0/cancel`.

In cases where the Identity Provider receives the BankID error code `ALREADY_IN_PROGRESS` in response to an Auth- or Sign-call the Identity Provider MAY display a warning to the user that someone may have initiated a BankID operation using their personal identity number[1]. If this warning is displayed, it is RECOMMENDED that the second level status code `http://id.elegnamnden.se/status/1.0/possibleFraud` is included in the error response message posted back to the Service Provider.

> [1]: There have been reports where fraudsters remotely try to convince people of using their Mobile BankID to log in to a service. In these cases, the fraudster initiates a BankID authentication prior to the person he tries to trick into logging in to the service, and is waiting for the user to enter his or hers personal code, thus authenticating the fraudsters session.

6. Metadata

This section extends section 2 of [EidProfile] with requirements specific for BankID.

6.1. Service Providers

As stated in section 5.1, "Attribute Release Rules", a Service Provider may request additional attributes, other than those implicitly requested via the use of service entity categories, by declaring requested attributes under the `<md:AttributeConsumingService>` element.

```
<md:AttributeConsumingService index="0" isDefault="true" xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
  <md:ServiceName xmlns:xml="http://www.w3.org/XML/1998/namespace" xml:lang="sv">E-myndigheten</md:ServiceName>
  <md:ServiceName xmlns:xml="http://www.w3.org/XML/1998/namespace" xml:lang="en">The e-Authority</md:ServiceName>
  ...
  <md:RequestedAttribute Name="urn:oid:1.2.752.201.3.2" isRequired="false"/>
  <md:RequestedAttribute Name="urn:oid:1.2.752.201.3.13" isRequired="false"/>
</md:AttributeConsumingService>
```

Example of how a Service Provider declares that it wishes to receive the transactionIdentifier and authServerSignature attributes in assertions.

A Service Provider requesting an attribute that is not supported by all Identity Providers that it may communicate with MUST NOT set the `isRequired` attribute of the `<md:RequestedAttribute>` element to `true`.

It is RECOMMENDED that Service Providers communicating with BankID Identity Providers include the `transactionIdentifier` attribute as a requested attribute.

As described in section 1.4, "Relying Party Configuration, a Service Provider may declare an entity category telling the BankID Identity Provider that it wishes QR codes to be used instead of prompting for the user personal identity number.

```
<mdattr:EntityAttributes xmlns:mdattr="urn:oasis:names:tc:SAML:metadata:attribute">
  <saml:Attribute Name="http://macedir.org/entity-category"
                  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
    <saml:AttributeValue xsi:type="xs:string">
      http://id.swedenconnect.se/general-ec/1.0/secure-authenticator-binding
    </saml:AttributeValue>
    ...
  </saml:Attribute>
</mdattr:EntityAttributes>
```

Example of how a Service Provider declares the secure authenticator binding entity category. This means, for a BankID Identity Provider, that QR codes should be used to initiate BankID operations.

6.2. Identity Providers

A BankID Identity Provider MUST 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`.

A BankID Identity Provider SHOULD declare the `<psc:RequestedPrincipalSelection>` element containing the attribute name for personalIdentityNumber (`urn:oid:1.2.752.29.4.13`) and include it in its metadata entry as described in section 2.1.3 of

[EidProfile].

Using this extension the Identity Provider announces that the requestor should send the personal identity number in the authentication request if this is known to the requestor. In this way, the Identity Provider does not have to prompt the user for the personal identity number for the use cases where this is required.

```
...
<md:IDPSSODescriptor WantAuthnRequestsSigned="true"
     protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
  <md:Extensions>
    <mdui:UIInfo xmlns:mdui="urn:oasis:names:tc:SAML:metadata:ui">
      ...
    </mdui:UIInfo>
    <psc:RequestedPrincipalSelection
         xmlns:psc="http://id.swedenconnect.se/authn/1.0/principal-selection/ns">
      <psc:MatchValue Name="urn:oid:1.2.752.29.4.13" />
    </psc:RequestedPrincipalSelection>
  </md:Extensions>
  <md:KeyDescriptor use="signing">
    ...
```

Example of how the Identity Provider declares the `RequestedPrincipalSelection` *extension in its metadata.*

A BankID Identity Provider supporting the BankID QR code feature MUST declare this in its metadata using the `http://id.swedenconnect.se/general-ec/1.0/secure-authenticator-binding` entity category.

6.3. Signature Services

It is RECOMMENDED that a Signature Service explicitly requires release of the `userSignature` attribute (`urn:oid:1.2.752.201.3.11`) in assertions. The reason for this is that the BankID-signature may then be part of the assertion that is included in the resulting signature created by the Signature Service giving a non-repudiation proof of the BankID signature process.

```
<md:AttributeConsumingService index="0" isDefault="true" xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
  <md:ServiceName xmlns:xml="http://www.w3.org/XML/1998/namespace" xml:lang="sv">E-myndighetens underskriftstjänst</md:ServiceName>
  <md:ServiceName xmlns:xml="http://www.w3.org/XML/1998/namespace" xml:lang="en">The e-Authority's Signing Service</md:ServiceName>
  ...
  <md:RequestedAttribute Name="urn:oid:1.2.752.201.3.11" isRequired="false"/>
</md:AttributeConsumingService>
```

Example of how the userSignature attribute is explicitly required.

7. References

[RFC2119]

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

[SAML2Core]

> OASIS Standard, Assertions and Protocols 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.

[SAML2MetaAttr]

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

[BankID_Spec]

> BankID Relying Party Guidelines, version 3.2.2.
>
> *Check www.bankid.com/rp/info for lastest version.*

[EidProfile]

> Deployment Profile for the Swedish eID Framework.

[EidAttributes]

> Attribute Specification for the Swedish eID Framework.

[EidEntCat]

> Entity Categories for the Swedish eID Framework.

[PrincipalSel]

> Principal Selection in SAML Authentication Requests.

8. Changes between versions

Changes between version 1.1 and 1.2:

- Section 1.4, "Relying Party Configuration", was added. The section describes how a Service Provider, using the `secure-authenticator-binding` entity category, may request that QR codes are used to initiate BankID operations (which is a more secure process).
- Section 3.3 was renamed from "Prompting for Personal Identity Number" to "Mobile BankID on another Device" and its contents was updated to add requirements of the use of QR codes.
- Since the BankID API now includes a cancel-method section 3.4, "Cancelling an Operation", was updated to require usage of this method if the user cancels the operation.
- Sections 3.3 and 4.2.2 were updated with new recommendations and requirements for the `<psc:PrincipalSelection>` extension.
- Section 6.2 was updated with a requirement that a BankID Identity Provider should include the `<psc:RequestedPrincipalSelection>` element in its metadata.
- The recommendation in section 3.2 concerning handling of non default browsers was updated.
- Section 4.2 was updated with a requirement that ensures that only the intended user sees a sign message.

Changes between version 1.0 and 1.1:

- Section 3.4, "Cancelling an Operation" was extended with a suggestion of how to avoid dangling sessions after user cancel.
- The profile now references the BankID Relying Party Guidelines that makes use of JSON.