My understanding of SAML Authentication in Alma

by Laurence Lockton

SAML is a protocol that can be used for accessing Alma (and Primo, Leganto, Esploro, etc.) by signing into your institution's Single Sign-on system, instead of having separate credentials in the Ex Libris identity service. This is often referred to as **federated login**. In SAML speak, Alma is the **Service Provider** (SP, or **Relying Party**) and your institution's single sign-on system is the **Identity Provider** (IdP). Universities mostly use Shibboleth software or the OpenAthens service as their IdP. Microsoft Azure Active Directory can also be used as an IdP with Alma.

This is how it works for me, to the best of my understanding.

I start by going to https://bath.alma.exlibrisgroup.com/mng/login?auth=SAML

Alma tells my browser to redirect to my institution's Shibboleth IdP with a URL that contains a SAML authentication request. This is an XML document which has been compressed and encoded in the SAMLRequest parameter of the URL, for example:

https://auth.bath.ac.uk/idp/profile/SAML2/Redirect/SSO?**SAMLRequest**=jZLBctowEIZfxaM7lpHbxtFgMnIcpszQluIkh94UewFNZUnVSjR5+xozTMklw1XS/vvtp53dvfY6OYBHZU1JpmlGEjCt7ZTZleTpcTEpyN18hrLXjosY9mYDfyJgSIY6g3y8KEn0hluJCrmRPSAPLW/EtxVnacadt8G2VpNEIIIPQ6N7azD24BvwB9XC02ZVkn0IDjmlLzLsU6l7mcKrVi9e4c7b6NLW9rQ3O+o6/CpNp2Fld8qQpB5glJFh5D+HyIE0PSW1afxNVefowLFVGugRjNENdMpDG2jT/CDJwvoWxvFKspUagSTLuiRi+kUURSXqhbiZMvZJ1JkoHliVV0VV3T6wz9K5bDq8xbVEVAf4X40YYWkwSBNKwjKWT7J8kmePWcHzW57fpFme/yLJ89n8YIqcPPOx1l8I/tivPFsl82sd6qO6Gb3odmrNHP8+5C/rtdWqfbtAYNd/stb2770HGQYbwUcY7fYyfBxwPFHdZDs+5cFLgwpMIEmzPiL9jFKrrQJ//aKMQxI6P835fnvn/wA=&**RelayState**=mng@@44BAT_INST

There is also RelayState parameter in the URL, which is typically used to store the URL for the final destination the user wants to get to after they've been granted access to a website. Alma just needs to know which application and institution the user is accessing, when the response comes back.

We can decode the SAMLRequest to see the XML. It contains some details of the Alma SP, an identifier for the request and a time stamp, because it can only be used for a limited time.

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    AssertionConsumerServiceURL="https://bath.alma.exlibrisgroup.com/mng/pdsHandleLogin"
    Destination="https://auth.bath.ac.uk/idp/profile/SAML2/Redirect/SSO"
    ForceAuthn="false" ID="A773B353E651818DC0EA2C7263D1B507Eapp01"
    IsPassive="false" IssueInstant="2023-03-30T08:47:12.723Z"
    Version="2.0">
 <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://bath.alma.exlibrisgroup.com/mng/login</saml:Issuer>
 <saml2p:NameIDPolicy xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol"
    AllowCreate="true"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"
    SPNameQualifier="https://bath.alma.exlibrisgroup.com/mng/login">
 </saml2p:NameIDPolicy>
</samlp:AuthnRequest>
```

My IdP asks me to sign in, if I'm not already, and returns a SAML response which my browser sends to the **AssertionConsumerServiceURL** given in the request:

https://bath.alma.exlibrisgroup.com/mng/pdsHandleLogin

The SAML response is sent in the body of an HTTP POST request, along with the RelayState parameter mng@@44BAT_INST.

Again, you can decode the SAML response to reveal a much larger XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2p:Response xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol" Destination="https://bath.alma.exlibrisgroup.com/mng/pdsHandleLogin"
ID="_591c6045ede9eeb4349b725205a139b9" InResponseTo="A773B353E651818DC0EA2C7263D1B507Eapp01" IssueInstant="2023-03-30T08:47:13.291Z" Version="2.0">
 <saml2:Issuer xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">https://idp.bath.ac.uk/shibboleth</saml2:Issuer>
 <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
   <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"></ds:CanonicalizationMethod>
   <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"></ds:SignatureMethod>
   <ds:Reference URI="#_591c6045ede9eeb4349b725205a139b9">
    <ds:Transforms>
     <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"></ds:Transform>
```

This contains user attributes, including an identifier which Alma uses to find my user record. The **assertions** in the SAML response, which include the user attributes, have been encrypted by the IdP using Alma's public key, so they can't be seen in your browser. Alma decrypts these using its private key. The message is also "signed" by the IdP using its private key, and Alma verifies the signature using the IdP's public key. More on that to follow.

Configuration

Before this will work, the Alma SP and the institution's IdP need some metadata about each other.

The SAML Integration Profile in Alma is where you enter the metadata for your institution's IdP, and configure how to handle the authentication requests. Also, confusingly in the middle of the page is where you export the metadata for the Alma SP to give to the IdP.

You can either enter the various details of the IdP into the SAML integration profile, or import the IdP's metadata file, which is formatted in XML, and Alma should extract the details from that.

A screenshot of the SAML Integration Profile in my institution's Alma is shown at the end of this document.

IdP issuer is better known as the **entity ID** of the IdP, its unique identifier.

IdP login URL specifies the how the SAML request will be sent from Alma to the IdP (the binding), in this case by HTTP redirect, and is matched to the **Destination** field in a SAML request.

User ID location is to tell Alma where to find the user identifier, that it will use to look up your user record, in the SAML response. If **User ID is an attribute element** you then have to provide the name of the attribute. In my configuration urn:oid:1.3.6.1.4.1.5923.1.1.1.6 is the name of the attribute better known as eduPersonPrincipalName, which for me is lislgl@bath.ac.uk and has to match one the identifiers in my Alma user record.

My institution's Shibboleth IdP had to be configured to release this identifier to the Alma SP, because by default it doesn't release any personally identifiable information.

The **IdP certificate**, which contains the public key used to verify the signature in the SAML response, can be entered or uploaded further down.

As mentioned, before that section is where you export the metadata file for the Alma SP to give to the administrator of your institution's IdP. You have to choose which certificate will be included in the metadata. That certificate contains the public key used by the IdP to encrypt the assertions in the SAML response. This is to provide extra security in case they contain personally identifiable information.

Certificate expiry

Certificates usually have an expiry date as a security measure. Ex Libris adds a new certificate to Alma from time to time, with an expiry date a couple of years in the future. When the certificate for a SP needs to be replaced, it has to be sent to the IdPs it interoperates with, and vice versa. This is normally done by sending out a replacement metadata file.

When the IdP certificate is due for renewal, you can enter the new certificate in advance of the certificate being switched over in the IdP, which is why there are two places to enter IdP certificates. When receiving a SAML response, Alma will try the public key from one certificate and if that fails, try the other. That way you can avoid downtime after the certificate is replaced in the IdP.

Note that only one SP certificate can be in selected in the SAML integration profile, because Alma can only send one SAML request to the IdP. Instead, you could create a second SAML integration profile configured with the new certificate. However, users would need to use a different URL which specifies the second integration profile code, like this:

https://bath.alma.exlibrisgroup.com/mng/login?auth=SAML&idpCode=SAML_Profile2

So that wouldn't really help, except that when the Alma SP certificate is changed in the IdP, you could use that method to get access to Alma and then make that the **Default SAML profile** so it works for everyone else when they go to the regular link.

Federations

An alternative to exchanging metadata to form a bilateral trust agreement between an IdP and a SP is for them to be included in a federation, such as the UK Access Management Federation, OpenAthens federation, Edugate in Ireland and InCommon in the US.

The federation sets some rules, so it's like a circle of trust, and recommendations, for example around the use of personally identifiable information. By default, IdPs in the UK Access Management Federation should only release the category of user for authorization purposes, e.g. staff@bath.ac.uk (the eduPersonScopedAffiliation attribute) and an opaque, persistent identifier to allow the SP to recognise the same user accessing again without knowing any personal details (eduPersonTargetedId). If the SP needs any personally identifiable information, the IdP has to be configured to release this specifically for that SP. As mentioned, Alma needs a user identifier to be released that it can use to lookup the user record.

The federation also maintains a central file of all the member IdP and SP metadata. Typically, all the IdPs and SPs in the federation "consume" this metadata file on a regular basis. So when your library subscribes to a resource on a content platform you've never used before, if the content platform's SP is in the federation, usually you don't need your IdP administrator to do anything because the IdP already knows about it.

Alma doesn't consume any federation metadata. This means that you do have to configure the details of your IdP in the SAML integration profile. However, you can ask Ex Libris to register your Alma SP in a federation, so that your IdP will automatically receive the metadata. The advantage of this is that when the Alma certificate is due to expire and needs replacing, you can ask Ex Libris to send the new metadata file to the federation. The IdP will automatically receive that when it next consumes the federation metadata file, and so you don't have to coordinate the switch with your IdP administrator. This is useful for me, because the single person who administers my institution's

IdP is very hard to pin down. But if you are the IdP administrator as well as the Alma administrator, or sit next to them, there's perhaps no advantage to having your Alma SP in a federation.

Self-registration

If you want to use self-registration, also known at auto-provisioning or just-in-time (JIT) user creation, your IdP has to be configured to release additional user attributes such as names, email address, department, etc., and set the **Mapping of assertion fields to Alma fields**. Alternatively the XSL configuration file can be used to do more elaborate mapping, for when the attribute values in the SAML Request don't neatly correlate to the user record field values in Alma.

General Information | Actions | Contact Info

SAML DEFINITIONS

Metadata upload method: Metadata link / Metadata upload

Metadata file link: [ ] Populate Profile

Default SAML profile: [x] ForceAuthn: [ ]

IdP issuer *: https://idp.bath.ac.uk/shibboleth

IdP login URL *: https://auth.bath.ac.uk/idp/profile/SAML2/Redirect/SSO

User ID location *: User ID is in an Attribute element

User ID attribute name *: urn:oid:1.3.6.1.4.1.5923.1.1.1.6

IdP logout URL: https://auth.bath.ac.uk/logout

IdP single logout service: [ ]

Sign single logout requests: [ ] Apply SHA1 signature (Default SHA2): [x]

Alma metadata file version *: Version 1 | Expiration date: 07 April 2023, Signed by: Self Signed, Signature algorithm: sha1RSA — Generate Metadata File

Service Provider metadata

IDP Certificate 1

Thumbprint -

Expiry date -

Uploaded on 08/03/2016

Download certificate — Remove certificate

Certificate upload method: Free-text certificate / JKS file / Certificate file

Enter certificate text: [ ]

IDP Certificate 2

Certificate upload method: Free-text certificate / JKS file / Certificate file

Enter certificate text: [ ] No certificate file uploaded

SELF REGISTRATION

Active: Active / Inactive

User Group: Patron not found

Resource sharing library: Inter-library loans

Statistical Category: Other dept

Update user upon login: [ ]

Recreate user roles upon login: [ ]

Mapping of assertion fields to Alma fields

XSL configuration file: [ ] Save