

IdP 構築・運用手順書 (Ver1.2)

1. 概要

本書は IdP の構築手順、および運用方法を説明したものです。

[1] IdP の機能

まず、IdP の動作について簡単に説明します。

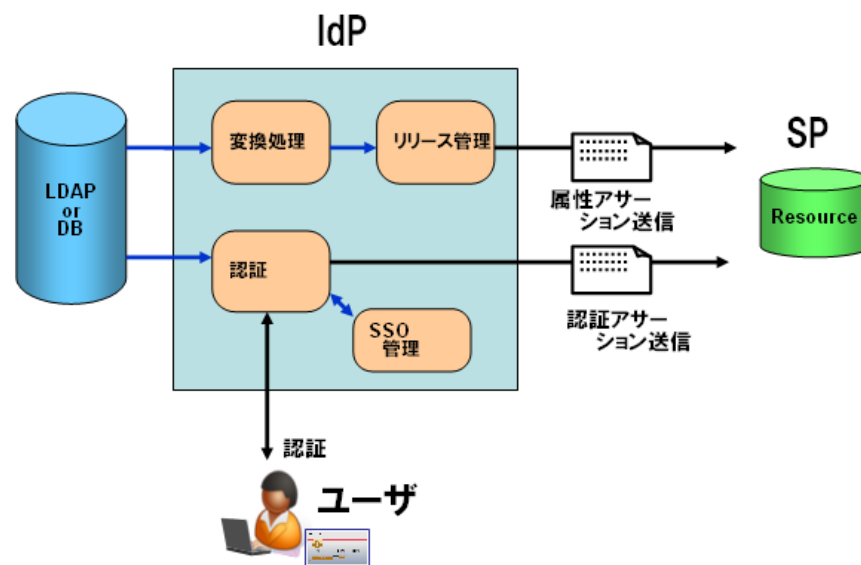


図 1 IdP の機能ブロック

図 1 IdP の機能ブロックは、IdP の機能を単純化したブロックで示しています。IdP は SP からの要求を受けて、以下の 2 つの動作を行います。

- ・ ユーザを認証する
- ・ ユーザの属性を安全に SP に送信する

■ 認証

ユーザが SP にアクセスすると、SP は IdP にリダイレクトを行います。IdP はこれを受けてユーザの認証を行います。認証方式としては、ID / パスワード認証や、クライアント証明書による認証等の認証方式が設定可能です。

また、IdP はユーザの Cookie を確認して、既に認証済みである場合は、2 回目以降のユーザに対する認証は行わず、シングルサインオン機能を実現します。

■ 属性の安全な送信

SP が要求する属性を、LDAP もしくは DB から取得して、SP に送信します。この際、下記

を行います。

- ・ LDAP に格納されている情報を元にして、SP が要求する属性とするために、名称の変更や、ドメインの追加といった変換を行います（図 1 の変換機能）
- ・ SP に送信して良いかどうか、ポリシーを確認します。各属性が送信可能である場合、SAML2.0 に準拠して安全に属性を送信します（図 1 のリリース機能）

以下の章では、IdP の構築手順を示すとともに、上記機能の設定方法、および、これらの機能を用いて IdP を運用するための方法について説明します。

[2] 構築方式について

IdP の構築にあたっては、以下の 2 つの方式のいずれかを採択することができます。

■ VMWare イメージを利用した構築

NII が提供する “ OS から shibboleth(IdP)までインストールされた ” システムを利用する方式です。貴学では、貴学のサーバに VMWareServer をインストールし、その上にこのシステムイメージを稼動することで利用できます。

■ 自分で IdP をインストールする場合の構築手順

貴学にて、貴学のサーバに OS から shibboleth(IdP)までインストール・設定を行い、構築する方式です。

本書では、上記 2 方式の手順を明記するとともに、構築後の基本的な設定方法や運用方法も記載しています。

2 . VMWare イメージを利用した構築手順

[1] 前提条件

(1) NII で動作検証した環境

- ・ ホスト OS : CentOS5.1
VMware コンソールを使用するので、Xwindow System をインストールしてください。
- ・ VMwareServer : VMware-server-1.0.4-56528.i386.rpm

(2) 配布する VM イメージの初期情報

■ VMwareServer での設定

- ・ ゲスト OS : 「 LINUX 」 - 「 Other Linux 2.6.x kernel 」
- ・ ネットワーク接続 : ブリッジ
- ・ ディスクサイズ : 4 GB
- ・ メモリサイズ : 2 5 6 MB

■ (ゲスト)OS での設定

- ・ (ゲスト)OS : CentOS 5.2 (Apache HTTP Server 2.2.3-11)
- ・ root パスワード : passwd
- ・ ホスト名 : upkishib11.nii.ac.jp
- ・ i p アドレス : 192.168.0.1
- ・ インストールソフトウェア :

開発	開発ツール (オプションパッケージは全て無し)
	開発ライブラリ (オプションパッケージは全て無し)
サーバ	Web サーバー (オプションパッケージは HTTP のみ)
	ネットワークサーバー (オプションパッケージは ldap Server のみ)
ベースシステム	ベースのみチェック (オプションはデフォルト)
X Window	なし

- ・追加インストール

SUN JDK 1.6、Apache Tomcat 6.0、openldap-2.3.27-8

- ・Firewall Configuration :

Security Level	Disabled
SELinux	Disabled

- ・認証設定：デフォルト(MD5,Shadow)
- ・サービス設定： ip6tables, iptables 停止。その他はデフォルト。

■ LDAP の初期設定

- ・suffix： o= test_o, dc=ac, c=JP
- ・rootdn： cn= olmgr, o= test_o, dc=ac, c=JP
- ・rootpw： csildap
- ・初期構成

uid	userPass word	eduPersonPrincipalName	ou	eduPerson Affiliation
testuid1	testpw1	test_eppn_1	science	faculty
testuid2	testpw2	test_eppn_2	economic	staff
testuid3	testpw3	test_eppn_3	technology	student

■ shibboleth インストールディレクトリ

/opt/shibboleth-idp-2.0.0

[2] VMwareServer をインストールする

■ ダウンロード URL

<http://www.vmware.com/jp/products/server/>

■ ダウンロードプロダクト

NII での動作検証環境では、以下のプロダクトをダウンロードし、インストールしました。

- ・VMware Server for Linux (VMware Server 本体)
- ・Management Interface (VMware 管理インタフェース)
- ・VMware Server Linux client package (VMware の仮想マシンコンソール)

■ インストール手順

http://www.vmware.com/jp/pdf/server_admin_manual.pdf

貴学の環境によっては、gcc、kernel-devel、kernel-headers、libXtst の追加インストールが必要となる場合があります。

インストールにはシリアルナンバーの入力を要します。

シリアルナンバーの取得は、以下の URL でアカウント登録をしてください。

<http://register.vmware.com/content/registration.html>

[3] VM イメージをダウンロードする

■ ダウンロード URL

以下の UPKI イニシアティブのサイトの「SSO 実証実験のリポジトリ」からダウンロードしてください。

<https://upki-portal.nii.ac.jp/SSO/Repository>

サイトへのアクセスには、UPKI イニシアティブ会員への登録が必要となります。

[4] VMwareServer に VM イメージを登録し、起動する。

当該サーバのホスト OS 上の「/var/lib/vmware/Virtual Machines」配下でダウンロードした VM イメージを格納し、解凍します。

```
#cd /var/lib/vmware/"Virtual Machines"  
#ls  
upkishibIdPv1.0.tar.gz  
#tar -zxvf upkishibIdPv1.0.tar.gz
```

upkishibIdP ディレクトリが作成され、その配下には以下の 5 つのファイルが作成されたことを確認します。

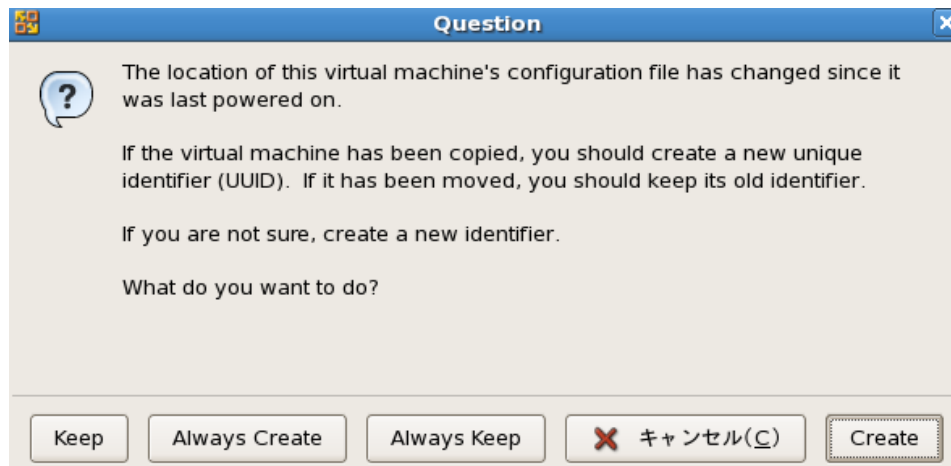
```
#ls /var/lib/vmware/"Virtual Machines"  
upkishibIdPv1.0.tar.gz  upkishibIdPv1.0  
# ls /var/lib/vmware/"Virtual Machines"/upkishibIdPv1.0  
upkishibIdP.vmx  upkishibIdP.vmdk  upkishibIdP.flat.vmdk  upkishibIdP.vmsd  nvram
```

upkishibIdP.vmx の権限を変更します。

```
# chmod 754 /var/lib/vmware/"Virtual Machines"/upkishibIdPv1.0/upkishibIdP.vmx
```

ホスト OS の X Window から「VMwareServer Console」を起動し、menu バーより
「File」 - 「Open」を選択し
「/var/lib/vmware/"Virtual Machines"/upkishibIdPv1.0/upkishibIdP.vmx」
を指定します。
「Power on this Virtul machine」をクリックします。

次のようなダイアログが表示されるので、「Create」を選択します。



仮想マシンを移動またはコピーした後に初めて仮想マシンをパワーオンすると、新しい UUID を生成するか聞いています。「Create」を選択することにより、MAC アドレスが新たに生成されます。

IdP がインストールされたゲスト OS (CentOS5.2) が起動します。

[5] ゲスト OS にログインする

- root の初期パスワードは passwd です。
- ログイン後、パスワードを変更してください。

```
#passwd
```

[6] ip アドレス、ホスト名を変更する

- 配布時は以下のように初期設定されていますので、貴学の環境に基づき変更してください。
 - ・ ip アドレス : 192.168.0.1
 - ・ ホスト名 : upkishib11.nii.ac.jp

- 変更箇所は以下の通りです。

```
/etc/sysconfig/network-scripts/ifcfg-eth0
```

```
DEVICE=eth0
ONBOOT=yes
IPADDR=192.168.0.1 ←ip アドレス
NETMASK=255.255.255.0 ←サブネットマスク
GATEWAY=192.168.0.254 ←ゲートウェイ
NETWORK=192.168.0.0 ←ネットワークアドレス
( 中略 )
```

```
/etc/sysconfig/network
```

```
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=upkishib11.nii.ac.jp ←ホスト名
```

/etc/resolv.conf

```
search nii.ac.jp ← ローカルドメイン名
nameserver 192.168.0.2 ← ネームサーバ
```

/etc/httpd/conf/httpd.conf

```
( 中略 )
ServerName upkishib11.nii.ac.jp:80 ← ホスト名
( 中略 )
```

/etc/httpd/conf.d/ssl.conf

```
( 中略 )
ServerName upkishib11.nii.ac.jp:443 ← ホスト名
( 中略 )
<VirtualHost _default_:443>
( 中略 )
ProxyPass /idp/ ajp://localhost:8009/idp/ ← 記述されているか確認
( 中略 )
</VirtualHost>
```

/opt/shibboleth-idp-2.0.0/metadata/idp-metadata.xml

(中略)

<EntityDescriptor entityID="https://upkishib11.nii.ac.jp/idp/shibboleth"> ← ホスト名

<IDPSSODescriptor protocolSupportEnumeration="urn:mace:shibboleth:1.0

urn:oasis:names:tc:SAML:2.0:protocol">

<KeyDescriptor>

<ds:KeyInfo>

<ds:X509Data>

<ds:X509Certificate>

MIIDOzCCAiOgAwIBAgIUdTJ6oiEccCjrtDyDaeBXTIRpfPcwDQYJKoZIhvcNAQEF

BQAwHzEdMBsGA1UEAxMUdXBraXNoaWlzM5uaWkuYWMuYWwHcNMDgwNzA4MTAz

NTQwWhcNMjgwNzA4MTAzNTQwWjAfmR0wGwYDVQQDEExR1cGtpc2hpYjExLm5paS5h

Yy5qcDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJlkO5kZI2Tz7tPg

1HVVhwv7g7bCYNL7Zx11IeNwDyd1ZluRAfmSzTNP67YPSMPLaA4upkYM51JmHgskZ

GvbjNqfma+imNV0/R7J0LL6ucSgL++ax25XKJmViCspQpgYPiFsavoF6JwciRdwk

zeUkhJo0zZWZ1rSxeAevAuWJf9hDwelCyry5u2ZNljDNLFl6uzpPETv3DSMxwevs

tHggag9917DSnH4ZhiBhXL3pd+g0qyw0ouuew9wtizZ6KpTtTlI3InuTM6KiCG5M

Iyv7HVc9KtwkVAooF/LMPP9ofkZeuqzpc8T6Wg+zaUUslKhEDhon4Zb/rt9tS3vB

JFpYBS8CAwEAAANvMG0wTAYDVR0RBEUwQ4IUdXBraXNoaWlzM5uaWkuYWMuYWwHcNMDgwNzA4MTAz

K2h0dHBzOi8vdXBraXNoaWlzM5uaWkuYWMuYWwHcNMDgwNzA4MTAz

VR00BBYEFCoPX1gOojzXICTZT7l73KcHkJSHMA0GCSqGSIb3DQEBBQUAA4IBAQA

GnudDV3eqTNLZPGH8zJWHCT8Az7CtG40aINRjzirbZI+r4X7Zuq5ZLv+n9EJ6rbd

xRW6bIx9YTLKcLvzXx0ZM4fy6RFyJ+8qCDTXig0DgVpng66xfJBi7ahjG1Jgn6

xXwdYFE50zLC3qwrZ9kykXCy2ELHfb3Z/g1o9fZZy7gjn77m1tDFWcs4M3NFCfL

zKbGNi+5a05w/wLkxpEaP8NPTHkbN3E+EXQDik7QOQgJ0+JEUYLAP06HTGGCs5i

YU+cTQ5QSGjfsSwcZQt6ljQUzlyhKOAwnazbrRGVfCVlwoY10hkpmGMSb4Jjxo6E

61psWSAHlehx6L2F9Eat

</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</KeyDescriptor>

<NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>

<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>

<SingleSignOnService Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"

Location="https://upkishib11.nii.ac.jp/idp/profile/Shibboleth/SSO"/> ← ホスト名

```

<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
    Location="https://upkishib11.nii.ac.jp/idp/profile/SAML2/POST/SSO" /> ←ホスト名
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
    Location="https://upkishib11.nii.ac.jp/idp/profile/SAML2/Redirect/SSO" /> ←ホスト名
</IDPSSODescriptor>
<Organization>
    <OrganizationName xml:lang="en">Test IdP</OrganizationName> ←組織名
    <OrganizationDisplayName xml:lang="en">Test IdP</OrganizationDisplayName> ←組織表示名
    <OrganizationURL xml:lang="en">http://YourHomePage/</OrganizationURL> ←組織 URL
</Organization>
<ContactPerson contactType="technical">
    <GivenName>YourGivenName</GivenName>
    <SurName>YourSurName</SurName> ←管理者名
    <EmailAddress>YourEmailAddress</EmailAddress> ←管理者の e-mail アドレス
</ContactPerson>
</EntityDescriptor>
( 中略 )

```

/opt/shibboleth-idp-2.0.0/conf/relying-party.xml

```

( 中略 )
<AnonymousRelyingParty provider="https://upkishib11.nii.ac.jp/idp/shibboleth" /> ←ホスト名
<DefaultRelyingParty provider="https://upkishib11.nii.ac.jp/idp/shibboleth" ←ホスト名
    defaultSigningCredentialRef="IdPCredential">
( 中略 )

```

- 新しいホスト名と i p アドレスを DNS に登録してください。
- 変更後のホスト名をヘルプデスク (upki-sso-help@nii.ac.jp) へ通知してください。

ヘルプデスクでは、SP・DS に貴学の IdP を登録し、これにより接続テストが可能となります。

[7] 時刻同期を設定、確認する

- ntp サービスを用い、貴学環境の ntp サーバと時刻同期をしてください。
Shibboleth では、通信するサーバ間の時刻のずれが約 5 分を越えるとエラーになります。
- VMwareServer のゲスト OS では、システムクロックが著しくずれますが、ハードウェアクロックのずれは少ないので、NII での検証では、以下の設定を施し安定稼働させています。

【設定例】

```
# crontab -l  
  
*/3 * * * * /sbin/clock -hctosys ← 3 分毎に clock コマンドでシステムクロックをハードウェアクロック  
                                     に合わせる  
  
10 0-23/1 * * * * ntpdate (ntp サーバ) && /sbin/clock -w ← 毎時 10 分に ntpdate コマンドで ntp サーバと  
                                                             同期し、ハードウェアクロックも合わせる
```

[8] セキュリティを設定、確認する

貴学のセキュリティポリシーに準拠し、サーバのセキュリティの設定・確認をしてください。

[9] サーバをリブートする

```
#reboot
```

お使いの環境によっては、極稀にサーバ起動時に tomcat サービスで”failed”が表示される場合があります。これは依存するサービスの起動に待ちが発生するためで、”failed”が表示されてもリトライされて正常起動するケースが高いです。

”failed”でも以降の [10] を実施し、それでもエラーとなる場合はヘルプデスク (upki-ss0-help@nii.ac.jp) へ連絡してください。

[1 0] テストアカウントで接続確認する

SP にアクセスする

現段階で用意されている SP は、以下の Plone サイトがあります。

<https://upkishib1.nii.ac.jp/>

Plone 画面が表示されたら、画面右端の「ログイン」をクリックし、「Shibboleth login」の「UPKI IdP_01」を選択、Shibboleth Identity Provider Login 画面から以下のアカウントでログインします。

id	password
testuid1	testpw1
testuid2	testpw2
testuid3	testpw3

[1 1] サーバ証明書を申請、登録する

「7 .シングルサインオン実証実験用 サーバ証明書の取得方法について」を参考に、サーバ証明書を申請します。

証明書の交付までには数日を要するので、お早めに申請してください。

入手したサーバ証明書をもとに、以下のファイルに設定してください。

■ /etc/httpd/conf.d/ssl.conf

```
( 中略 )
SSLCertificateFile /etc/pki/tls/certs/server.crt ←サーバ証明書の格納先
( 中略 )
SSLCertificateKeyFile /etc/pki/tls/private/server.key ←秘密鍵の格納先
( 中略 )
#SSLCACertificateFile /etc/pki/tls/certs/ca-bundle.crt ←コメントアウト
SSLCACertificatePath /opt/shibboleth-idp-2.0.0/credentials/CA ←中間 CA 証明書の格納先
```

/opt/shibboleth-idp-2.0.0/credentials/CA ディレクトリが無い場合は作成してください。このディレクトリには、ファイル名をハッシュ値とした中間 CA 証明書を配置します。

詳しくは UPKI「サーバ証明書プロジェクト」<https://upki-portal.nii.ac.jp/cerpj> を参照してください。

■ /opt/shibboleth-idp-2.0.0/conf/relying-party.xml

```
( 中略 )
<security:Credential id="IdPCredential" xsi:type="security:X509Filesystem">
  <security:PrivateKey>/opt/shibboleth-idp-2.0.0/credentials/server.key ←ssl.confと同一のものを
  </security:PrivateKey>                                         左記のパスにも格納
  <security:Certificate>/opt/shibboleth-idp-2.0.0/credentials/server.crt ←ssl.confと同一のものを
  </security:Certificate>                                         左記のパスにも格納
</security:Credential>
( 中略 )
```

以下の UPKI イニシアティブのサイトの「SSO 実証実験のリポジトリ」から IdP 用メタデータテンプレートをダウンロードし、必要な項目を変更します。

<https://upki-portal.nii.ac.jp/SSO/Repository>

サイトへのアクセスには、UPKI イニシアティブ会員への登録が必要となります。

ダウンロードしたメタデータテンプレートを下記のように変更してください。

(中略)

<EntityDescriptor entityID="https://upkishib11.nii.ac.jp/idp/shibboleth"> ← ホスト名

<IDPSSODescriptor protocolSupportEnumeration="urn:mace:shibboleth:1.0
urn:oasis:names:tc:SAML:2.0:protocol">

<KeyDescriptor>

<ds:KeyInfo>

<ds:X509Data>

<ds:X509Certificate>

MIIDOzCCAIOgAwIBAgIUdTJ3oiEccCjrtDyDaeBXTlRpfPcwDQYJKoZIhvcNAQEF
BQAwHzEdMBsGA1UEAxMUdXBraXNoaWxMS5uaWkuYWUanAwHhcNMDgwNzA4MTAz
NTQwWhcNMjgwNzA4MTAzNTQwWjAfmR0wGwYDVQQDEXR1cGtpc2hpYjExLm5paS5h
Yy5qcDCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBABJlkO5kZI2Tz7tPg
1HVVhwv7g7bCYNL7Zx11IeNwDyd1ZluRAfmSzTNP67YPSMPLA4A4upkYM51JmHgskZ
GvbJNqfma+imNV0/R7J0LL6ucSgL++ax25XKJmViCspQpgYPIFsavoF6JwciRdwk
zeUkhJo0zZWZ1rSxeAevAuWJf9hDwelCyry5u2ZNIjDNLFl6uzpPETv3DSMxwevs
tHggag9917DSnH4ZhiBhXL3pd+g0qyw0ouuew9wtizZ6KpTtTlI3InuTM6KiCG5M
Iyv7HVc9KtwkVAooF/LMPP9ofkZeuqzpc8T6Wg+zaUUslKhEDhon4Zb/r9tS3vB
JFpYBS8CAwEAAANvMG0wTAYDVR0RBEUwQ4IUdXBraXNoaWxMS5uaWkuYWUanAwHhcNMDgwNzA4MTAz
K2h0dHBzOi8vdXBraXNoaWxMS5uaWkuYWUanAwWRwL3NoaWJib2xlDGgwHQYD
VR0OBByEFCoPX1gOojzXlCTZT7l73KcHkJSHMA0GCsGSIb3DQEBBQUAA4IBAQA
WGNudDV3eqTNLZPGH8zJWHCT8Az7CtG40aINRJzibrZi+r4X7Zuq5ZLv+n9EJ6rbd
xRWb6bIx9YTLKcLvzXx0ZM4fy6RFyJ+8qCDTXig0qDgVpng66xfJB7ahjGIJgn6
xXwdYFE50zLC3qwrZ9kykXCy2ELLHfb3Z/g1o9fZZy7gjn77m1tDfWcs4M3NFCFL
zKbGni+5a05w/wLkxpEaP8NPTHkbN3E+EXQDik7QOqGJ0+JEUYLAP06HTGGCs5i
YU+cTQ5QsgjfsSwcZQt6ljQUzlyhKOAwnazbrRGVfCVlwoYl0hkpmGMSb4Jjxo6E
61psWSAHlehx6L2F9Eat

</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</KeyDescriptor>

<NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>

<NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>

<SingleSignOnService Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"

Location="https://upkishib11.nii.ac.jp/idp/profile/Shibboleth/SSO"/> ← ホスト名

入手した証明書に変更

```

<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
    Location="https://upkishib11.nii.ac.jp/idp/profile/SAML2/POST/SSO" /> ← ホスト名
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
    Location="https://upkishib11.nii.ac.jp/idp/profile/SAML2/Redirect/SSO" /> ← ホスト名
</IDPSSODescriptor>
<Organization>
    <OrganizationName xml:lang="en">Test IdP</OrganizationName> ← 組織名
    <OrganizationDisplayName xml:lang="en">Test IdP</OrganizationDisplayName> ← 組織表示名
    <OrganizationURL xml:lang="en">http://YourHomePage/</OrganizationURL> ← 組織 URL
</Organization>
<ContactPerson contactType="technical">
    <GivenName>YourGivenName</GivenName>
    <SurName>YourSurName</SurName> ← 管理者名
    <EmailAddress>YourEmailAddress</EmailAddress> ← 管理者の e-mail アドレス
</ContactPerson>
</EntityDescriptor>
( 中略 )

```

完成した新しい IdP 用のメタデータを、ヘルプデスク (upki-sso-help@nii.ac.jp) へ送付してください。

ヘルプデスクでは、送付していただいたメタデータを SP・DS に登録するとともに共用メタデータを更新します。

詳しくは、以下のサイトを参照してください。

<https://upki-portal.nii.ac.jp/SSO/Repository>

サイトへのアクセスには、UPKI イニシアティブ会員への登録が必要となります。

同様に以下の UPKI イニシアティブのサイトの「SSO 実証実験のリポジトリ」から upki-fed-metadata.xml をダウンロードし、ファイル名を idp-metadata.xml に変更して、/opt/shibboleth-idp-2.0.0/metadata と差し替えてください。

<https://upki-portal.nii.ac.jp/SSO/Repository>

サイトへのアクセスには、UPKI イニシアティブ会員への登録が必要となります。

3 . 貴学にて IdP をインストールする場合の構築手順

[1] shibboleth (IdP version2.0)の動作要件

- ・ Apache HTTP Server 2.2 以上
- ・ Apache Tomcat 5.5.25 以上
- ・ Java 5 以上

(ただし、CentOS に付属する Gnu Java は利用できません。 Sun の Java を利用してください。)

[2] OS をインストールする

OS での設定

- ・ OS : CentOS 5.1

インストーラでインストールするもの。

Web サーバー (HTTP のみ)

Open Ldap

その他のパッケージは必要に応じてインストールしてください。

ただし、Java 開発と Tomcat は後の手順で別にインストールします。

- ・ ネットワーク設定

環境に合わせ、ホスト名・ネットワーク・セキュリティを設定して下さい。

SP では shibd サービスが通信を行います。

DNS へ登録する

新しいホスト名と i p アドレスを DNS に登録してください

時刻同期を設定する

ntp サービスを用い、貴学環境の ntp サーバと時刻同期をしてください。

Shibboleth では、通信するサーバ間の時刻のずれが約 5 分を越えるとエラーになります。

[3] jdk6、tomcat6 をインストールする

tomcat5-5.5.17-8 の削除

tomcat5-5.5.25 以前のバージョンが入っている場合は、削除してください。

jdk 6.0 のインストール

<http://java.sun.com/javase/downloads/index.jsp> よりダウンロードした jdk-6u5-linux-i586-rpm.bin を適当なフォルダに置いて、以下のコマンドを実行してください。

```
chmod a+x jdk-6u5-linux-i586-rpm.bin
./jdk-6u5-linux-i586-rpm.bin
```

tomcat6.0.16 のインストール

<http://tomcat.apache.org/download-60.cgi> よりダウンロードした apache-tomcat-6.0.16.tar.gz を /usr/java に置いて、以下のコマンドを実行してください。

```
cd /usr/java
tar zxvf apache-tomcat-6.0.16.tar.gz
ln -s apache-tomcat-6.0.16 /usr/java/tomcat
```

jsvc などを用いて、自動起動させると便利です。

ソースファイルを展開し、make で jsvc を作成した後、\$CATALINA_HOME/bin にコピーします。起動用スクリプトをコピーします。

```
cd /usr/java/tomcat/bin
tar xzvf jsvc.tar.gz
cd jsvc-src
./configure
make
cp jsvc ..
cp native/Tomcat5.sh /etc/rc.d/init.d/tomcat
```

/etc/rc.d/init.d/tomcat の先頭にコメントを追加することにより chkconfig コマンドが利用できます。

```
# chkconfig: - 86 15
# description: Tomcat
# processname: tomcat
```

また、/etc/rc.d/init.d/tomcat ファイル中の以下の環境変数も変更が必要です。

```
JAVA_HOME  
CATALINA_HOME  
DAEMON_HOME  
CATALINA_BASE
```

設定例

```
JAVA_HOME=/usr/java/default  
CATALINA_HOME=/usr/local/tomcat  
DAEMON_HOME=$CATALINA_HOME  
CATALINA_BASE=$CATALINA_HOME
```

tomcat を実行するユーザ “tomcat” を作成した場合には

```
TOMCAT_USER=tomcat
```

も設定します。

他に、“tomcat”ユーザがログファイルを出力できるよう、ディレクトリの所有者を変更します。(シンボリックリンク先を変更するため最後の “/” が必要です)

```
chwon -R tomcat: /usr/java/tomcat/
```

自動起動の設定 (オプションは マイナス ‘-’ が2つ必要です)

```
chkconfig --add tomcat  
chkconfig --level 345 tomcat on
```

/etc/profile に下記を追加してください。

```
JAVA_HOME=/usr/java/default  
MANPATH=$MANPATH:$JAVA_HOME/man  
CATALINA_HOME=/usr/java/tomcat  
TOMCAT_HOME=$CATALINA_HOME  
PATH=$JAVA_HOME/bin:$CATALINA_HOME/bin:$PATH  
export PATH JAVA_HOME CATALINA_HOME
```

Apache ant インストール

IdP のインストールには Apache ant が必要です。

もしインストールされていない場合にはインストールして下さい。

httpd の設定

/httpd/conf/httpd.conf の修正

```
(省略)  
ServerName upkishib11.nii.ac.jp:80 ←ホスト名  
(省略)
```

/etc/httpd/conf.d/ssl.conf の修正

```
(省略)  
  
SSLCertificateFile /opt/shibboleth-idp-2.0.0/credentials/idp.crt  
SSLCertificateKeyFile /opt/shibboleth-idp-2.0.0/credentials/idp.key  
#SSLCACertificateFile /etc/pki/tls/certs/ca-bundle.crt ←コメントアウト  
SSLCACertificatePath /opt/shibboleth-idp-2.0.0/credentials/CA  
  
(省略)  
ServerName upkishib11.nii.ac.jp:443 ←ホスト名  
(省略)  
<VirtualHost _default_:443>  
(省略)  
ProxyPass /idp/ ajp://localhost:8009/idp/ ←追加  
(省略)  
</VirtualHost>
```

/opt/shibboleth-idp-2.0.0/credentials/CA ディレクトリが無い場合は作成してください。このディレクトリには、ファイル名をハッシュ値とした中間 CA 証明書を配置します。

詳しくは UPKI「サーバ証明書プロジェクト」<https://upki-portal.nii.ac.jp/cerpj> を参照してください。

/usr/java/tomcat/conf/server.xml の修正

- 必要に応じて Connector port="8080"をコメントアウトしてください。

```
<!--  
    <Connector port="8080" protocol="HTTP/1.1"  
        connectionTimeout="20000"  
        redirectPort="8443" />  
-->
```

- Connector port="8009"に以下のように追加してください。

```
<Conector port="8009"  
    protocol="AJP/1.3" redirectPort="8443" />  
↓  
<Conector port="8009"  
    protocol="AJP/1.3" redirectPort="8443" enableLookups="false"  
    request.tomcatAuthentication="false" address="127.0.0.1" />
```

[3] openldap の設定

追加のスキーマファイル

eduperson スキーマを以下からダウンロードし、解凍したものを
/etc/openldap/schema/ に置いてください。

<http://www.educause.edu/eduperson/>

「eduPerson (200412)」ボタン下の「Platform-Specific eduPerson LDIFs」を
クリックしてください。

OpenLDAP 用の eduPerson(200412) をダウンロードします。

解凍時のファイルの拡張子が*.ldif となっているので、*.schema に変更してください。

ldap のデフォルト設定

/etc/openldap/slapd.conf を変更します。

```
( 省略 )  
include /etc/openldap/schema/eduperson-200412.schema ←追加  
  
suffix " o= test_o, dc=ac, c=JP " ←suffix  
rootdn " cn= olmgr, o= test_o, dc=ac, c=JP " ←rootdn  
rootpw {CRYPT}ijFYncSNctBYg ←暗号化 したものを記載
```

ここで設定したパスワードは IdP の設定ファイルにも記述します。(後述)

暗号化の例 : 「csildap」というパスワードを暗号化する

```
#slappasswd -h {crypt} -s csildap
```

```
{CRYPT}ijFYncSNctBYg
```

←これを slapd.conf の rootpw に記載

ldap のテストデータ作成

以下のサンプルを基に、テスト用データを作成し、ldap へ登録します。

Shibboleth を利用した ID/パスワードでの認証に使用される ID は uid 、パスワードは userPassword になります。

- test.ldif ファイル作成

```
dn: o=test_o,dc=ac,c=JP
objectClass: organization
o: test_o
```

```
dn: ou=science, o=test_o, dc=ac, c=JP
objectClass: organizationalUnit
ou: science
```

```
dn: ou=economic, o=test_o, dc=ac, c=JP
objectClass: organizationalUnit
ou: economic
```

```
dn: ou=technology, o=test_o, dc=ac, c=JP
objectClass: organizationalUnit
ou: technology
```

```
dn: eduPersonPrincipalName=test_eppn_1, ou=science, o=test_o, dc=ac, c=JP
objectClass: eduPerson
objectClass: inetOrgPerson
eduPersonPrincipalName: test_eppn_1
ou: science
sn: test_sn_1
cn: test_cn_1
uid: testuid1
userPassword: testpw1
eduPersonAffiliation: faculty
```

- ldap への登録

```
#ldapadd -x -h localhost -D "cn=olmgr,o=test_o,dc=ac,c=JP" -w csildap -f test.ldif
```

[4] shibboleth のインストール

shibboleth-idp-2.0.0-bin.zip のダウンロード

<http://shibboleth.internet2.edu/downloads/shibboleth/idp/2.0/> から
shibboleth-idp-2.0.0-bin.zip をダウンロードします。

インストール

shibboleth-idp-2.0.0-bin.zip を展開し、インストールします。

```
# unzip shibboleth-idp-2.0.0-bin.zip
# cd identityprovider
# bash ant.sh
```

いくつかの質問に答えながら、インストールを行います。

途中で入力するパスワードはデフォルトで作成されるキーストアファイルのパスワードとなります。

Java の設定

shibboleth-idp-2.0.0-bin.zip を展開した identityprovider/lib/ にある
shib-jce-1.0.jar を \$JAVA_HOME/jre/lib/ext にコピーします。

さらに、 \$JAVA_HOME/jre/lib/security/java.security ファイルに以下を追加します。

```
security.provider.9=edu.internet2.middleware.shibboleth.DelegateToApplicationProvider
```

番号:9 は既に記載されている番号に合わせて順番にして下さい。

Tomcat の設定

shibboleth-idp-2.0.0-bin.zip を解凍した identityprovider/endorsed にある
4 つの jar ファイルを \$CATALINA_HOME/endorsed ディレクトリを作成して
そこへコピーします。

(Tomcat5.5 の場合は \$CATALINA_HOME/common/endorsed ディレクトリ)

```
xalan-2.7.1-serializer.jar
xalan-2.7.1.jar
xerces-2.9.1-xercesImpl.jar
xerces-2.9.1-xml-apis.jar
```

これらの jar ファイルが有効となるよう、Tomcat 起動スクリプトを変更します。
/etc/rc.d/init.d/tomcat を作成していた場合は、以下の追加となります。

```
CATALINA_OPTS="-Djava.endorsed.dirs=${CATALINA_HOME}/endorsed "
```

上記例ではデフォルトで記述されている

“-Djava.library.path=/home/jfclere/jakarta-tomcat-connectors/jni/native/.libs”
を使用していないため記述を削除しています。

tomcat を、“tomcat”ユーザで実行する場合は、ログファイルを出力できるようディレクトリの所有者を変更します。

```
chwon -R tomcat: /opt/shibboleth-idp-2.0.0/logs
```

idp.war の配置

/opt/shibboleth-idp-2.0.0/war/idp.war ファイルを、
\${CATALINA_HOME}/webapps ディレクトリにコピーします。

[4] shibboleth の設定

デフォルトでは shibboleth は /opt/shibboleth-idp-2.0.0 ディレクトリにインストールされます。変更する各設定ファイルは /opt/shibboleth-idp-2.0.0/conf にあります。

・ idp-metadata.xml ファイルの設定

同様に以下の UPKI イニシアティブのサイトの「SSO 実証実験のリポジトリ」から
upki-fed-metadata.xml をダウンロードし、ファイル名を idp-metadata.xml に変更
して、/opt/shibboleth-idp-2.0.0/metadata と差し替えてください。

<https://upki-portal.nii.ac.jp/SSO/Repository>

サイトへのアクセスには、UPKI イニシアティブ会員への登録が必要となります。

・ relying-party.xml ファイルの確認

ホスト名が正しいことを確認します。

ダウンロードしたメタデータの記述を追加します。

```
<!-- ===== -->
<!--      Relying Party Configurations      -->
<!-- ===== -->
<AnonymousRelyingParty provider="https://upkishib11.nii.ac.jp/idp/shibboleth" />
<DefaultRelyingParty provider="https://upkishib11.nii.ac.jp/idp/shibboleth"
    defaultSigningCredentialRef="IdPCredential">
(省略)
<!-- ===== -->
<!--      Metadata Configuration      -->
<!-- ===== -->
<!-- MetadataProvider the combining other MetadataProviders -->
<MetadataProvider id="ShibbolethMetadata" xsi:type="ChainingMetadataProvider"
    xmlns="urn:mace:shibboleth:2.0:metadata">
(省略)
    <MetadataProvider id="FSMD" xsi:type="FilesystemMetadataProvider"
        xmlns="urn:mace:shibboleth:2.0:metadata"
        metadataFile="/opt/shibboleth-idp-2.0.0/metadata/idp-metadata.xml"
        maintainExpiredMetadata="true" />
</MetadataProvider>
```

・ attribute-resolver.xml ファイルの変更

デフォルトの /opt/shibboleth-idp-2.0.0/conf/attribute-resolver.xml を元に、以下の手順に従い変更を行ってください。

「organizationName」を検索し、場所を特定してください。(行番号は参考です)

```
162 -->      ←コメント終了を追加して、以下の “organizationName”, “organizationalUnit” の値を公開します
163
164      <resolver:AttributeDefinition id="organizationName" xsi:type="Simple"
                                xmlns="urn:mace:shibboleth:2.0:resolver:ad"
165      sourceAttributeID="o">
166      <resolver:Dependency ref="staticAttributes" />      ← myLDAP を staticAttributes に変更
167
168      <resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
169      name="urn:mace:dir:attribute-def:o" />
170
171      <resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
172      name="urn:oid:2.5.4.10" friendlyName="o" />
173  </resolver:AttributeDefinition>
174
175  <resolver:AttributeDefinition id="organizationalUnit" xsi:type="Simple"
                                xmlns="urn:mace:shibboleth:2.0:resolver:ad"
176  sourceAttributeID="ou">
177  <resolver:Dependency ref="myLDAP" />
178
179  <resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
180  name="urn:mace:dir:attribute-def:ou" />
181
182  <resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
183  name="urn:oid:2.5.4.11" friendlyName="ou" />
184  </resolver:AttributeDefinition>
185
186 <!--      ←コメント開始を追加して、以下をコメントアウトします
```

「eduPerson attributes」を検索し、場所を特定してください。(行番号は参考です)

```
323  <!-- Schema: eduPerson attributes -->
324  <!-- -->      ← コメント終了を追加して、以下を有効とします
325  <resolver:AttributeDefinition id="eduPersonAffiliation" xsi:type="Simple"
                                xmlns="urn:mace:shibboleth:2.0:resolver:ad"
326  sourceAttributeID="eduPersonAffiliation">
327  <!--      <resolver:Dependency ref="staticAttributes" /> -->      ← コメントアウトし
328  <resolver:Dependency ref="myLDAP" />      ← myLDAP に変更
329
330  <resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
331  name="urn:mace:dir:attribute-def:eduPersonAffiliation" />
332
333  <resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
334  name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1" friendlyName="eduPersonAffiliation" />
335  </resolver:AttributeDefinition>
336
337  <!--      ← コメント開始を追加して、以下をコメントアウトします
```

「eduPersonPrincipalName」を検索し、場所を特定してください。(行番号は参考です)

```
403 ---> ←コメント終了を追加して、以下を有効とします
404           ↓ eduPersonPrincipalName から変更します
405     <resolver:AttributeDefinition id="principalName" xsi:type="Scoped"
406           xmlns="urn:mace:shibboleth:2.0:resolver:ad"
407           scope="nii.ac.jp" sourceAttributeID="eduPersonPrincipalName"> ←ドメイン名を変更
408     <resolver:Dependency ref="myLDAP" />           ↑ uid から変更します
409     <resolver:AttributeEncoder xsi:type="SAML1ScopedString"
410           xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
411           name="urn:mace:dir:attribute-def:eduPersonPrincipalName" />
412     <resolver:AttributeEncoder xsi:type="SAML2ScopedString"
413           xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
414           name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" friendlyName="eduPersonPrincipalName" />
415   </resolver:AttributeDefinition>
416   <!-- ←コメント開始を追加して、以下をコメントアウトします
```

「Data Connectors」を検索し、場所を特定してください。(行番号は参考です)

```
460   <!-- ===== -->
461   <!--      Data Connectors      -->
462   <!-- ===== -->
463
464   <!-- Example Static Connector -->
465   <!-- ---> ←コメント終了を追加して、以下を有効とします
466   <resolver:DataConnector id="staticAttributes" xsi:type="Static"
467           xmlns="urn:mace:shibboleth:2.0:resolver:dc">
468     <Attribute id="o"> ←追加
469       <Value>test_o</Value>
470     </Attribute>
471   <!-- ←コメントアウト
472     <Attribute id="eduPersonAffiliation">
473       <Value>member</Value>
474     </Attribute>
475     <Attribute id="eduPersonEntitlement">
476       <Value>urn:example.org:entitlement:entitlement1</Value>
477       <Value>urn:mace:dir:entitlement:common-lib-terms</Value>
478     </Attribute>
479   </resolver:DataConnector>
480   <!-- ---> ←コメントアウト
```

「LDAP Connector」を検索し、場所を特定してください。(行番号は参考です)

```
498   <!-- Example LDAP Connector -->
499   <!-- ---> ←コメント終了を追加して、以下を有効とします
500   <resolver:DataConnector id="myLDAP" xsi:type="LDAPDirectory"
501           xmlns="urn:mace:shibboleth:2.0:resolver:dc"
502           ldapURL="ldap://localhost" baseDN="o=test_o,dc=ac,c=JP" principal="cn=olmgr,o=test_o,dc=ac,c=JP"
503           principalCredential="csildap"> ←LDAP のパスワードを設定
504     <FilterTemplate>
505       <![CDATA[
506         (uid=$requestContext.principalName)
507       ]]>
508     </FilterTemplate>
509   </resolver:DataConnector>
510   <!-- ---> ←コメントアウト
```

・ attribute-filter.xml ファイルの変更

デフォルトの /opt/shobboleth-2.0.0/conf/attribute-filter を元に、以下の変更を行ってください。

「transientId」を検索し、場所を特定してください。(行番号は参考です)

```
18 <!-- Release the transient ID to anyone -->
19 <AttributeFilterPolicy id="releaseTransientIdToAnyone">
20   <PolicyRequirementRule xsi:type="basic:ANY" />
21
22   <AttributeRule attributeID="transientId">
23     <PermitValueRule xsi:type="basic:ANY" />
24   </AttributeRule>
25
26   <AttributeRule attributeID="principalName"> ←追加
27     <PermitValueRule xsi:type="basic:ANY" />
28   </AttributeRule>
29
30   <AttributeRule attributeID="eduPersonAffiliation"> ←追加
31     <PermitValueRule xsi:type="basic:ANY" />
32   </AttributeRule>
33
34   <AttributeRule attributeID="organizationName"> ←追加
35     <PermitValueRule xsi:type="basic:ANY" />
36   </AttributeRule>
37
38   <AttributeRule attributeID="organizationalUnit"> ←追加
39     <PermitValueRule xsi:type="basic:ANY" />
40   </AttributeRule>
41
42 </AttributeFilterPolicy>
```

- handler.xml ファイルの変更

LDAP のデータを用いた ID/パスワード認証のために handler.xml ファイルを変更します。

```
<!-- Login Handlers -->
<!--                                     ← コメントアウトします
<LoginHandler xsi:type="RemoteUser">
    <AuthenticationMethod>
        urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</AuthenticationMethod>
    </LoginHandler>
-->

<!-- Username/password login handler -->    ← こちらを有効にします
<LoginHandler xsi:type="UsernamePassword"
    jaasConfigurationLocation="file:///opt/shibboleth-idp-2.0.0/conf/login.config">
    <AuthenticationMethod>
        urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
    </AuthenticationMethod>
</LoginHandler>
```

- login.config ファイルの変更

LDAP のデータを用いた ID/パスワード認証のために login.config ファイルを変更します。 各設定値は、ldap のデフォルト設定と同じ値とします。

```
ShibUserPassAuth {

// Example LDAP authentication
// See: https://spaces.internet2.edu/display/SHIB2/IdPAuthUserPass
edu.vt.middleware.ldap.jaas.LdapLoginModule required
    host="localhost"
    base="o=test_o,dc=ac,c=JP"
    ssl="false"
    userField="uid"
    subtreeSearch="true"
;    ← 最後の ; は必ず記述
```

[5] サーバ証明書を申請、登録する

「7 .シングルサインオン実証実験用 サーバ証明書の取得方法について」を参考に、サーバ証明書を申請します。

証明書の交付までには数日を要するので、お早めに申請してください。

入手したサーバ証明書をもとに、以下のファイルに設定してください。

■ /etc/httpd/conf.d/ssl.conf

```
( 省略 )
SSLCertificateFile /etc/pki/tls/certs/server.crt ←サーバ証明書の格納先
( 省略 )
SSLCertificateKeyFile /etc/pki/tls/private/server.key ←秘密鍵の格納先
( 省略 )
#SSLCACertificateFile /etc/pki/tls/certs/ca-bundle.crt ←コメントアウト
SSLCACertificatePath /opt/shibboleth-idp-2.0.0/credentials/CA ←中間 CA 証明書の格納先
```

/opt/shibboleth-idp-2.0.0/credentials/CA ディレクトリが無い場合は作成してください。このディレクトリには、ファイル名をハッシュ値とした中間 CA 証明書を配置します。

詳しくは UPKI「サーバ証明書プロジェクト」<https://upki-portal.nii.ac.jp/cerpj> を参照してください。

■ /opt/shibboleth-idp-2.0.0/conf/relying-party.xml

```
( 中略 )
<security:Credential id="IdPCredential" xsi:type="security:X509Filesystem">
  <security:PrivateKey>/opt/shibboleth-idp-2.0.0/credentials/server.key ←ssl.confと同一のものを
  </security:PrivateKey>                                         左記のパスにも格納
  <security:Certificate>/opt/shibboleth-idp-2.0.0/credentials/server.crt ←ssl.confと同一のものを
  </security:Certificate>                                         左記のパスにも格納
</security:Credential>
( 中略 )
```

以下の UPKI イニシアティブのサイトの「SSO 実証実験のリポジトリ」から IdP 用メタデータテンプレートをダウンロードし、必要な項目を変更します。

<https://upki-portal.nii.ac.jp/SSO/Repository>

サイトへのアクセスには、UPKI イニシアティブ会員への登録が必要となります。

ダウンロードしたメタデータテンプレートを下記のように変更してください。

(中略)

<EntityDescriptor entityID="https://**upkishib11.nii.ac.jp**/idp/shibboleth"> ← ホスト名

<IDPSSODescriptor protocolSupportEnumeration="urn:mace:shibboleth:1.0
urn:oasis:names:tc:SAML:2.0:protocol">

<KeyDescriptor>

<ds:KeyInfo>

<ds:X509Data>

<ds:X509Certificate>

MIIDOZCAiOgAwIBAgIUdTJ6oiEccCjrtDyDaeBXTlRpPcwDQYJKoZIhvcNAQEF
BQAwHzEdMBsGA1UEAxMUdXBraXNoaWxMS5uaWkuYWUuanAwHhcNMDgwNzA4MTAz
NTQwWhcNMjgwNzA4MTAzNTQwWjAfmR0wGwYDVQQDEExR1cGtpc2hpYjExLm5paS5h
Yy5qcDCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAJlkO5kZI2Tz7tPg
1HVVHwv7g7bCYNL7Zx11IeNwDyd1ZluRAfmSzTNP67YPSMPLaA4upkYM51JmHgskZ
GvbJNqfma+imNV0/R7J0LL6ucSgL++ax25XKJmViCspQpgYPiFsaVoF6JwciRdwk
zeUkhJo0zZWZ1rSxeAevAuWJf9hDwelCyry5u2ZNljDNLFI6uzpPETv3DSMxwevs
tHggag9917DSnH4ZhiBhXL3pd+g0qyw0ouuew9wtizZ6KpTtTl3InuTM6KiCG5M
Iyv7Hvc9KtwkVAooF/LMPP9ofkZeuqzpc8T6Wg+zaUUslKhEDhon4Zb/rt9tS3vB
JFpYBS8CAwEAAANvMG0wTAYDVR0RBGUwQ4IudXBraXNoaWxMS5uaWkuYWUuanCG
K2h0dHBzOi8vdXBraXNoaWxMS5uaWkuYWUuanAvaWRwL3NoaWJib2xldGgwHQYD
VR0OBByEFCoPX1gOojzXICTZT7173KcHkJSHMA0GCSqGSIb3DQEBBQUAA41BAQAW
GnudDV3eqTNLZPGH8zJWHCT8Az7CtG40aINRjzirbZI+r4X7Zuq5ZLv+n9EJ6rbd
xRWb6bIx9YTLKcLvzXx0ZM4fy6RFyJ+8qCDTXig0qDgVpng66xfJBj7ahjGIJgn6
xXwdYFE50zLC3qwrZ9kykXCy2ELLHfb3Z/g1o9fZZy7gjn77m1tDfWcs4M3NFCfL
zKbGNI+5a05w/wLkxpEaP8NPTHkbN3E+EXQDik7QQOqGJ0+JEUYLAP06HTGCGCs5i
YU+cTQ5QSgJfsSwcZQt6ljQUzlyhKOAwnazbrRGVfCVIwoY10hkpmGMSb4Jjxo6E
61psWSAHlehx6L2F9Eat

</ds:X509Certificate>

入手した証明書に変更

```

        </ds:X509Data>

        </ds:KeyInfo>

    </KeyDescriptor>

    <NameIDFormat>urn:mace:shibboleth:1.0:nameIdentifier</NameIDFormat>

    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>

    <SingleSignOnService Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"
        Location="https://upkishib11.nii.ac.jp/idp/profile/Shibboleth/SSO" /> ← ホスト名

    <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
        Location="https://upkishib11.nii.ac.jp/idp/profile/SAML2/POST/SSO" /> ← ホスト名

    <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
        Location="https://upkishib11.nii.ac.jp/idp/profile/SAML2/Redirect/SSO" /> ← ホスト名

</IDPSSODescriptor>

<Organization>

    <OrganizationName xml:lang="en">Test IdP</OrganizationName> ← 組織名

    <OrganizationDisplayName xml:lang="en">Test IdP</OrganizationDisplayName> ← 組織表示名

    <OrganizationURL xml:lang="en">http://YourHomePage/</OrganizationURL> ← 組織 URL

</Organization>

<ContactPerson contactType="technical">

    <GivenName>YourGivenName</GivenName>

    <SurName>YourSurName</SurName> ← 管理者名

    <EmailAddress>YourEmailAddress</EmailAddress> ← 管理者の e-mail アドレス

</ContactPerson>

</EntityDescriptor>

```

(中略)

完成した新しい IdP 用のメタデータを、ヘルプデスク (upki-sso-help@nii.ac.jp) へ送付してください。

ヘルプデスクでは、送付していただいたメタデータを SP・DS に登録するとともに共用メタデータを更新します。

詳しくは、以下のサイトを参照してください。

<https://upki-portal.nii.ac.jp/SSO/Repository>

サイトへのアクセスには、UPKI イニシアティブ会員への登録が必要となります。

[6] サービスを起動・停止方法

httpd の起動方法

```
service httpd start
```

tomcat の起動方法

```
service tomcat start      (jsvc を利用した場合)
```

httpd の停止方法

```
service httpd stop
```

tomcat の停止方法

```
service tomcat start      (jsvc を利用した場合)
```

[7] テストアカウントで接続確認する

SP にアクセスする

現段階で用意されている SP は、以下の Plone サイトがあります。

<https://upkishib1.nii.ac.jp/>

Plone 画面が表示されたら、画面右端の「ログイン」をクリックし、「Shibboleth login」の「UPKI IdP_01」を選択、Shibboleth Identity Provider Login 画面から以下のアカウントでログインします。

id	password
testuid1	testpw1
testuid2	testpw2
testuid3	testpw3

4 . 構築後の基本的な設定、変更方法

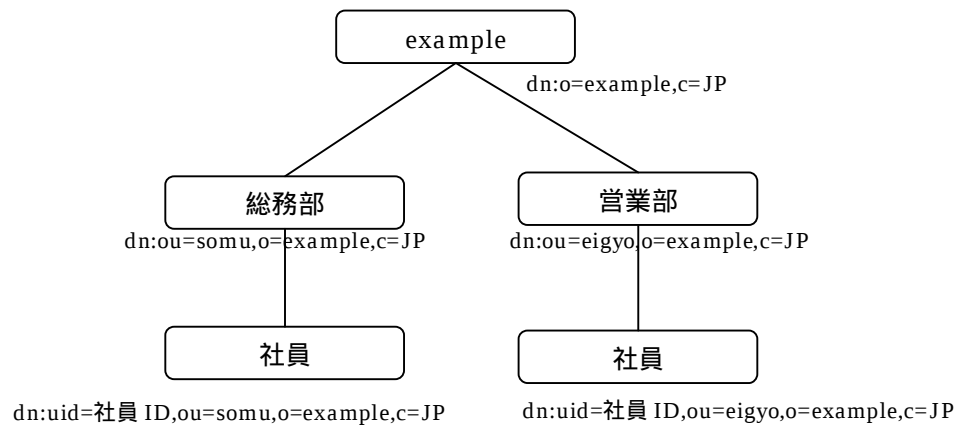
[1] LDAP の新規作成方法

既存 LDAP の削除

```
#ldapdelete -x -h localhost -D "cn=olmgr, o=test_o,dc=ac,c=JP" -w csildap
```

新しい LDAP のツリー構造の設計

以下の組織構造を新規作成する例を示します。



[2] 認証方法の変更、設定（証明書による認証）

1 章、あるいは、2 章で行った LDAP を利用した ID/パスワード認証の他に、様々な認証方法を利用することが可能です。以下では、クライアント証明書を利用した認証の設定方法を示します。

この例では、

- ・クライアント証明書を発行するキャンパス認証局の CA 証明書 = Camp-CA.crt
- ・クライアント証明書のサブジェクト”o”の値 = “ Test_University_A ”

として設定を行い、クライアント証明書が有効な証明書であり、かつ、上記の条件を満たす場合に認証を行う設定としています。

また、eduPersonPrincipalName をキーとして、クライアント証明書のサブジェクト “ CN”の値により LDAP から各属性を取得しています。そのため、クライアント証明書のサブジェクト”CN”の値を、LDAP の eduPersonPrincipalName に格納しておく必要があります。

- ・ /opt/shibboleth-idp-2.0.0/conf/handler.xml の変更（オリジナルに戻す）

クライアント証明書を用いた認証のために handler.xml ファイルを変更します。

```
<!-- Login Handlers -->
    ← “<!-- ” を削除して、こちらを有効にします

    <LoginHandler xsi:type="RemoteUser">
        <AuthenticationMethod>
            urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</AuthenticationMethod>
        </LoginHandler>
    ← “--> ” を削除して、こちらを有効にします

<!-- Username/password login handler -->
<!--      ← コメントアウトします
<LoginHandler xsi:type="UsernamePassword"
    jaasConfigurationLocation="file:///opt/shibboleth-idp-2.0.0/conf/login.config">
    <AuthenticationMethod>
        urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
    </AuthenticationMethod>
</LoginHandler>
-->      ← コメントアウトします
```

- /etc/httpd/conf.d/ssl.conf への追加

```
( 省略 )
<VirtualHost _default_:443>
( 省略 )
ProxyPass /idp ajp://localhost:8009/idp
<Location /idp/Authn/RemoteUser> ←追加
    SSLCACertificateFile /opt/shibboleth-idp-2.0.0/credentials/Camp-CA.crt ←追加
    SSLVerifyClient require ←追加
    SSLVerifyDepth 3 ←追加
    SSLRequireSSL ←追加
    SSOOptions +ExportCertData +StdEnvVars ←追加
    SSLUserName SSL_CLIENT_S_DN_CN ←追加
    SSLRequire %{SSL_CLIENT_S_DN_O} eq "Test_University_A" ←追加
</Location> ←追加
( 省略 )
</VirtualHost>
```

- /opt/shibboleth-idp-2.0.0/conf/attribute-resolver.xml の修正 1

また、クライアント証明書のサブジェクト “CN”の値を eduPersonPrincipalName に設定して、これをキーとして LDAP から属性を取得するため、下記の設定を行います。

「eduPersonPrincipalName」を検索し、場所を特定してください。(行番号は参考です)

```
404
405 <resolver:AttributeDefinition id="principalName" xsi:type="Scoped"
      xmlns="urn:mace:shibboleth:2.0:resolver:ad"
406     scope="nii.ac.jp" sourceAttributeID="eduPersonPrincipalName">
407     <resolver:Dependency ref="remoteUser" /> ←変更
408
409     <resolver:AttributeEncoder xsi:type="SAML1ScopedString"
      xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
410     name="urn:mace:dir:attribute-def:eduPersonPrincipalName" />
411
412     <resolver:AttributeEncoder xsi:type="SAML2ScopedString"
      xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
413     name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" friendlyName="eduPersonPrincipalName" />
414 </resolver:AttributeDefinition>
415
416 <resolver:AttributeDefinition xsi:type="PrincipalName" xmlns=
      "urn:mace:shibboleth:2.0:resolver:ad" id="remoteUser" /> ←変更
```

- /opt/shibboleth-idp-2.0.0/conf/attribute-resolver.xml の修正 2

「LDAP Connector」を検索し、場所を特定してください。(行番号は参考です)

```
498 <!-- Example LDAP Connector -->
499 <!-- -->
500 <resolver:DataConnector id="myLDAP" xsi:type="LDAPDirectory"
                                xmlns="urn:mace:shibboleth:2.0:resolver:dc"
501     ldapURL="ldap://localhost" baseDN="o=test_o,dc=ac,c=JP"
                                principal="cn=olmgr,o=test_o,dc=ac,c=JP"
502     principalCredential="csildap">
503   <FilterTemplate>
504     <![CDATA[
505       (eduPersonPrincipalName=$requestContext.principalName) ←変更
506     ]]>
507   </FilterTemplate>
508 </resolver:DataConnector>
509 <!-- -->
```

5 . 運用方法

[1] metadata の管理方法

メタデータは新規 IdP、新規 SP の追加や、既存 IdP、既存 SP の証明書の更新等により、常に更新されます。そのため、定期的にリポジトリから最新の共通メタデータをダウンロードして IdP のメタデータを更新してください。

また、証明書の更新等、IdP のメタデータに更新があった場合は、すみやかにヘルプデスクに送付してください。

[2] 新規 SP の登録方法

新規 SP の情報が入ったメタデータを IdP に更新するだけで、新規 SP と接続できるようになります。

・メタデータの更新

以下の UPKI イニシアティブのサイトの「SSO 実証実験のリポジトリ」から upki-fed-metadata.xml をダウンロードし、これを

/opt/shibboleth-idp-2.0.0/metadata/idp-metadata.xml に上書きしてください。

<https://upki-portal.nii.ac.jp/SSO/Repository>

基本的な接続を行うための貴学での設定は以上です。

ただし、接続のためには新規 SP と DS (新規 SP が利用する場合) に貴学の IdP 情報が登録されることが必要です。

[3] 属性管理（登録、変換、リリース方法）

各属性は 1 章、2 章で設定を行った通り、基本的には多くの属性を LDAP や DB から取得してリリースするための設定が既に入っており、これらのコメントを削除して有効化するだけで実行することができます。

また、属性の変換機能として、“@nii.ac.jp”といったドメインの付与、値の変換、固定値の割り当てや、スクリプトを利用した変換等が可能です。

さらに、リリース制御では、サイトとしてのポリシー、各個人のポリシーによる制御や、各 SP に対応したリリース制御等が可能です。

以下、[4] - [6] では、新たな ID や属性の追加、そのリリースの方法について説明します。

[4] ID の追加方法

Uid	userPassword	eduPersonPrincipalName	ou	eduPersonAffiliation
testuid4	testpw4	test_eppn_4	technology	student

以下の利用者を追加する例を示します。

ldif ファイル(sample1.ldif)の作成

```
# test_eppn_4, technology, test_o, ac, JP
dn: eduPersonPrincipalName=test_eppn_4,ou=technology,o=test_o,dc=ac,c=JP
objectClass: eduPerson
objectClass: inetOrgPerson
eduPersonPrincipalName: test_eppn_4
ou: technology
sn: test_sn_4
cn: test_cn_4
uid: testuid4
userPassword: testpw4
eduPersonAffiliation: student
```

上記 の ldif ファイルを用いた登録

```
#ldapadd -x -h localhost -D "cn=olmgr, o=test_o,dc=ac,c=JP" -w csildap -f example1.ldif
```

[5] 属性の追加方法

利用者に「displayName」属性を追加する例を示します。

ldif ファイル(sample2.ldif)の作成

```
dn: eduPersonPrincipalName=test_eppn_4,ou=technology,o=test_o,dc=ac,c=JP
changetype: modify
add : displayName
displayName:Test4
```

上記 の ldif ファイルを用いた登録

```
# ldapmodify -x -h localhost -D "cn=olmgr, o=test_o,dc=ac,c=JP" -w csildap -f example2.ldif
```

[6] 属性のリリース方法

[5] で追加した「displayName」属性を SP へリリースする例を示します。

スキーマの確認

- ・ /etc/openldap/schema 配下にスキーマファイルがあります。
- ・「displayName」属性は、/etc/openldap/schema/inetorgperson.schema にて以下のように定義されています。

```
(中略)
# displayName
# When displaying an entry, especially within a one-line summary list, it # is useful to be able to
identify a name to be used. Since other attri- # bute types such as 'cn' are multivalued, an
additional attribute type is # needed. Display name is defined for this purpose.
attributetype ( 2.16.840.1.113730.3.1.241
    NAME 'displayName'
    DESC 'RFC2798: preferred name to be used when displaying entries'
    EQUALITY caseIgnoreMatch
    SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
    SINGLE-VALUE )
(中略)
```

/opt/shibboleth-idp-2.0.0/conf/attribute-resolver.xml への登録

```
<AttributeResolver xmlns="urn:mace:shibboleth:2.0:resolver" xmlns:resolver="urn:mace:shibboleth:2.0:resolver"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:pc="urn:mace:shibboleth:2.0:resolver:pc"
  xmlns:ad="urn:mace:shibboleth:2.0:resolver:ad" xmlns:dc="urn:mace:shibboleth:2.0:resolver:dc"
  xmlns:enc="urn:mace:shibboleth:2.0:attribute:encoder" xmlns:sec="urn:mace:shibboleth:2.0:security"
  xsi:schemaLocation="urn:mace:shibboleth:2.0:resolver classpath:/schema/shibboleth-2.0-attribute-resolver.xsd
    urn:mace:shibboleth:2.0:resolver:pc classpath:/schema/shibboleth-2.0-attribute-resolver-pc.xsd
    urn:mace:shibboleth:2.0:resolver:ad classpath:/schema/shibboleth-2.0-attribute-resolver-ad.xsd
    urn:mace:shibboleth:2.0:resolver:dc classpath:/schema/shibboleth-2.0-attribute-resolver-dc.xsd
    urn:mace:shibboleth:2.0:attribute:encoder classpath:/schema/shibboleth-2.0-attribute-encoder.xsd
    urn:mace:shibboleth:2.0:security classpath:/schema/shibboleth-2.0-security.xsd">

  (中略)

  <resolver:AttributeDefinition id="displayName" xsi:type="Simple" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
    sourceAttributeID="displayName">
    <resolver:Dependency ref="myLDAP" />
    <resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
      name="urn:mace:dir:attribute-def:displayName" />
    <resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
      name="urn:oid:2.16.840.1.113730.3.1.241" ← Oid
      friendlyName="displayName" />
  </resolver:AttributeDefinition>

  (中略)
</AttributeResolver>
```

/opt/shibboleth-idp-2.0.0/conf/attribute-filter.xml への登録

```
(中略)
<!-- Release the transient ID to anyone -->
<AttributeFilterPolicy id="releaseTransientIdToAnyone">
  <PolicyRequirementRule xsi:type="basic:ANY" />
  (中略)
  <AttributeRule attributeID="displayName">
    <PermitValueRule xsi:type="basic:ANY" />
  </AttributeRule>
  (中略)
</AttributeFilterPolicy>
(中略)
```

追加

ldif ファイル(sample3.ldif)の作成

```
#TOP ツリー用
dn: o=example,c=JP
objectClass: organization
o: example
#部署ツリー用
dn: ou=somu,o=example,c=JP
objectClass: organizationalUnit
ou: somu
dn: ou=eigyo,o=example,c=JP
objectClass: organizationalUnit
ou: eigyo
#社員ツリー用
dn: uid=u001,ou=somu,o=example,c=JP
objectClass: inetOrgPerson
cn: takesi
sn: yamada
uid: u001
userPassword: p-yamada

dn: uid=u002,ou=eigyo,o=example,c=JP
objectClass: inetOrgPerson
cn: taro
sn: maeda
uid: u002
userPassword: p-maeda
```

/etc/openldap/slapd.conf の編集

```
( 中略 )
suffix      "o=exmample,c=JP"      ← suffix
rootdn      "cn=Manager,o=example,c=JP"  ← rootdn
rootpw      {SSHA}2ZbK+0LyV92py+vi67X80RNAKg066GXS  ←暗号化 したものを記載
( 中略 )
```

暗号化の例 : 「csildap2」というパスワードを暗号化する

```
#slappasswd -h {SSHA} -s csildap2  
80U6H/KA4fVlvC+6DzN73DTP/dEAgl76 ←これを slapd.conf の rootpw に記載
```

LDAP サービスの再起動

```
#service ldap restart
```

上記 の ldif ファイルを用いた登録

```
#ldapadd -x -h localhost -D "cn=Manager,o=example,c=JP " -w csildap2 -f example3.ldif
```

/opt/shibboleth-idp-2.0.0/conf/attribute-resolver.xml の編集

```
<resolver:DataConnector id="myLDAP" xsi:type="LDAPDirectory" xmlns="urn:mace:shibboleth:2.0:resolver:dc"  
  ldapURL="ldap://localhost" baseDN=" o=example,c=JP " principal=" cn=Manager,o=example,c=JP "  
  principalCredential="csildap2"> ←LDAP のパスワードを設定  
  <FilterTemplate>  
    <![CDATA[  
      (uid=$requestContext.principalName)  
    ]]>  
  </FilterTemplate>  
</resolver:DataConnector>
```

tomcat サービスの再起動

```
#service tomcat stop  
#service tomcat start
```

6. シングルサインオン実証実験用 サーバ証明書の取得方法について

実証実験で構築する IdP（認証サーバ）には、サーバ証明書の導入が必須となります。

また、本実証実験においては、UPKI「サーバ証明書発行・導入における啓発・評価研究プロジェクト」が発行するサーバ証明書を利用しますので、以下の手順に従い、サーバ証明書の申請を行ってください。

【サーバ証明書発行・導入における啓発・評価研究プロジェクト 参加予定機関の方】

(1)プロジェクトへの参加

最初に、本プロジェクトへの参加申請への参加申請が必要となります。詳細については、次のページをご覧ください。

- サーバ証明書発行・導入における啓発・評価研究プロジェクト概要・参加要領等

<https://upki-portal.nii.ac.jp/cerpi>

(2)サーバ証明書の発行

プロジェクト参加完了後にサーバ証明書の発行を行います。次の手続きに従って、サーバ証明書の発行を申請してください。なお、CSR 作成にあたっては、次頁の「CSR プロファイル」を適用してください。

- 新規サーバ証明書発行手続き

https://upki-portal.nii.ac.jp/cerpi/request_new

(3)サーバ証明書インストール

次の手順に従って、サーバ証明書を IdP にインストールしてください。

- サーバ証明書のインストール方法

<https://upki-portal.nii.ac.jp/cerpi/niiodcamanual-v1-0.pdf>

【サーバ証明書発行・導入における啓発・評価研究プロジェクト 参加機関の方】

次の手順に従って、新規に証明書の発行手続きおよびインストールを行ってください。

(1)新規サーバ証明書発行手続き

https://upki-portal.nii.ac.jp/cerpi/request_new

なお、CSR 作成にあたっては、次頁の「CSR プロファイル」を適用してください。

(2)サーバ証明書インストール方法

<https://upki-portal.nii.ac.jp/cerpi/niiodcamanual-v1-0.pdf>

【CSR プロファイル】

本実証実験で使用するサーバ証明書の CSR は以下の形式で作成してください。

基本領域		設定内容	補
Version		Version 1(0)	-
Subject	Country	C=JP (固定値)	1
	Locality	L=Academe (固定値)	1
	Organization	O="主体者組織名" * 機関毎に任意に指定 例) o= National Institute of Informatics	1
	Organizational Unit	OU="主体者組織単位名" * 証明書毎に任意に指定 例) ou= NII Open Domain CA	1
	commonName	CN="サーバ FQDN" * 証明書毎に任意に指定 例) cn=www.nii.ac.jp	1
SubjectPublicKeyInfo		主体者の公開鍵 1024 ビット以上 (ただし、例外を認める)	2
attrobites		原則 Null 値とする (ただし、例外を認める)	3
SignatureAlgorithm		SHA1 with RSAEncryption	
1. 上記指定以外の属性を利用する必要がある場合には事前相談すること。少なくとも ST (state or province name) 属性は使用しないこと。また、例えば加入者メールアドレスなど本プロジェクトの確認項目対象外の情報を含めないこと。 2. RSA1024bit 以上とする。鍵長 1024bit 未満の場合には事前に登録局へ相談すること。 3. 任意の属性を含めても構わないが、必ずしも証明書に反映されるわけではない。また、含めた属性によっては受理不能とし、当該属性を除いて証明書発行要求の再生成を登録局から求める場合がある。少なくとも SubjectAltName.rfc822Name 属性は使用しないこと。			

7 . 関連 URL

- UPKI プロジェクト (UPKI イニシアティブ)
<https://upki-portal.nii.ac.jp/>
- UPKI 認証連携基盤によるシングルサインオン実証実験
<https://upki-portal.nii.ac.jp/SSO>
- UPKI 認証連携基盤リポジトリ
<https://upki-portal.nii.ac.jp/SSO/Repository>
- Shibboleth プロジェクト
<http://shibboleth.internet2.edu/>
- Shibboleth2.0 Wiki (Shibboleth2.0 の構築、設定に関する公式サイト)
<https://spaces.internet2.edu/display/SHIB2/Home>
- Switch.aai (スイスのフェデレーション)
<http://www.switch.ch/aai/>
- InCommon (米国のフェデレーション)
<http://www.incommonfederation.org/>