2024年11月8日

TOTPを用いた多要素認証方式の導入

国立情報学研究所

教育研修事業

**目次**

[**1. はじめに 2**](#_heading=h.3znysh7)

[**2. 実習セミナーでは 3**](#_heading=h.1t3h5sf)

[**2.1. Shibboleth IdPの設定 3**](#_heading=h.gytru5ixshj)

[**2.2. TOTPシード作成 6**](#_heading=h.8khrjl3oigla)

[**2.3. LDAPの設定 6**](#_heading=h.33g2t75rfune)

[**3. 手順書 10**](#_heading=h.4d34og8)

[**4. 動作確認 11**](#_heading=h.kg5sa8txu84y)

[**4.1. TOTP対応アプリの設定** 11](#_heading=h.pmchlqhiuiyi)

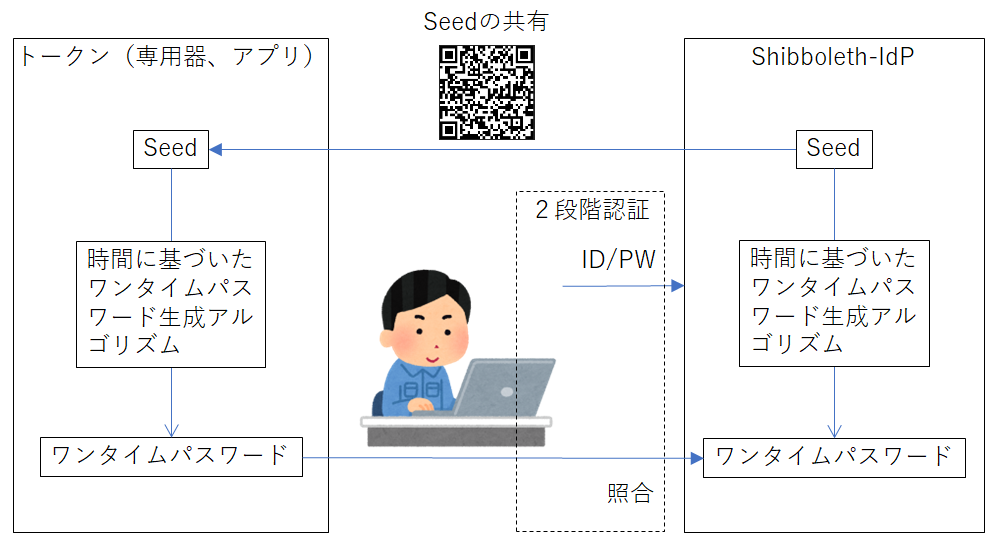
[**4.2. ログイン 15**](#_heading=h.vp2831byi8iz)

# **1. はじめに**

Shibboleth IdPでの認証において、Google Authenticatorなど、TOTP（Time-based One-Time Password）に対応したデバイス/アプリのワンタイムパスワードを必要とするように、TOTPプラグインを追加します。

TOTPは、一度しか利用できない「ワンタイムパスワード」の一種であり、時間に基づいた乱数からパスワードを生成します。生成されるパスワードは30秒から1分ほどで切り替わるため、不正アクセスなどのリスクを抑えられるのが特徴です。

動作確認にはTOTPに対応したデバイスもしくはアプリが必要です。



#### 図 TOTP認証の仕組み

# 

# **2. 実習セミナーでは**

以下のような設定で行います。

## **2.1. Shibboleth IdPの設定**

IdP 4.2 以降では、IdP バージョンでサポートされている最新のプラグイン バージョンをインストールできます。

* 以下のコマンドによりTOTPプラグインをインストールします。

|  |
| --- |
| # cd /opt/shibboleth-idp  # ./bin/plugin.sh -I net.shibboleth.idp.plugin.authn.totp [yN] y |

* 以下のコマンドによりシードを暗号化・復号化するためのAES鍵を作成して、キーストアに格納します。

|  |
| --- |
| # cd /opt/shibboleth-idp  # ./bin/seckeygen.sh --alias totp --count 1 --storefile credentials/totpsealer.jks --storepass password --versionfile credentials/totpsealer.kver |

* 先ほど作成したAES鍵をTOTP認証で使用するため、/opt/shibboleth-idp/conf/global.xmlの末尾（</beans>の上）に以下（赤字）を挿入します。

|  |
| --- |
| <bean id="totp.DataSealerKeyStrategy" lazy-init="true"  class="net.shibboleth.shared.security.impl.BasicKeystoreKeyStrategy"  p:keystoreType="JCEKS"  p:keystoreResource="%{idp.home}/credentials/totpsealer.jks"  p:keyVersionResource="%{idp.home}/credentials/totpsealer.kver"  p:keyAlias="totp"  p:keystorePassword="password"  p:keyPassword="password"  p:updateInterval="PT1H" />  <bean id="totp.DataSealer" lazy-init="true"  class="net.shibboleth.shared.security.DataSealer"  p:keyStrategy-ref="totp.DataSealerKeyStrategy" />  </beans> |

* TOTPシードを格納する属性を指定します。ここではbusinessCategoryを使用します。/opt/shibboleth-idp/conf/authn/authn.propertiesの末尾に以下を挿入します。

|  |
| --- |
| # Most actual setup via mfa-authn-config.xml  #### TOTP ####  idp.authn.TOTP.tokenSeedAttribute = businessCategory |

* TOTPシードを格納する属性を定義します。

/opt/shibboleth-idp/conf/attribute-resolver.xmlに以下を追加します。

|  |
| --- |
| </AttributeDefinition>  <AttributeDefinition xsi:type="Decrypted" id="businessCategory" dataSealerRef="totp.DataSealer">  <InputDataConnector ref="myLDAP" attributeNames="businessCategory"/>  </AttributeDefinition>    <!-- ===================== -->  <!-- Data Connectors -->  <!-- ====================== --> |

* 以下コマンドを実行して、MFAモジュールを有効化します。  
  ※活用編メニュー「セキュリティレベルを設定したSPに対する認証」が実施済みの環境であれば既にMFAモジュールが有効化されているため、本手順は不要です。

|  |
| --- |
| # /opt/shibboleth-idp/bin/module.sh -t idp.authn.MFA || /opt/shibboleth-idp/bin/module.sh -e idp.authn.MFA |

* /opt/shibboleth-idp/conf/authn/authn.propertiesにて、idp.authn.flowsにMFAフロー以外が指定されている場合はMFAフローを指定します。

|  |
| --- |
| idp.authn.flows = MFA |

* すべての認証動作をPassword認証→TOTP認証の順序とするため、/opt/shibboleth-idp/conf/authn/mfa-authn-config.xmlを以下のように設定します。

|  |
| --- |
| the final event, then the MFA process attempts to complete itself successfully.  ->  <util:map id="shibboleth.authn.MFA.TransitionMap">  <!-- Run authn/Password first. -->  <entry key="">  <bean parent="shibboleth.authn.MFA.Transition" p:nextFlow="authn/Password" />  </entry>    <!-- If that returns "proceed", run authn/TOTP next. -->  <entry key="authn/Password">  <bean parent="shibboleth.authn.MFA.Transition" p:nextFlow="authn/TOTP" />  </entry>    <!-- An implicit final rule will return whatever the second flow returns. -->  </util:map>  </beans> |

* /opt/shibboleth-idp/bin/build.shを実行し、WARファイル再作成およびデプロイします。

|  |
| --- |
| # /opt/shibboleth-idp/bin/build.sh |

* 最後にJettyの再起動を行います。

|  |
| --- |
| # systemctl restart jetty |

## 

## **2.2. TOTPシード作成**

以下の手順でTOTP対応アプリに設定するTOTPシードを作成します。

① 以下コマンドによりTOTPシードの作成を行います。

|  |
| --- |
| # cd /opt/shibboleth-idp  # ./bin/totpauth.sh --quiet --home . --issuer Shibboleth --account [テストユーザ名]  Seed: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  URL:\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  QR Code: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

※実習セミナー環境では、[テストユーザ名]として、test001, test002, test003が定義されています。

Seedの値は後ほど暗号化するために利用するので保存してください。

QR CodeのURLも、動作確認で利用するので同様に保存してください。

② 先ほど生成したSeedを以下コマンドにより暗号化します。

|  |
| --- |
| # cd /opt/shibboleth-idp  # ./bin/sealer.sh --quiet conf/global.xml enc [上で保存したSeed]  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*←暗号化されたSeed |

## **2.3. LDAPの設定**

テストユーザー用に暗号化されたSeed値をLDAPに登録します。

本手順書ではbusinessCategory属性に登録します。

LDAPに登録する時は、ldapsearch, ldapmodifyコマンドを使いますが、そのコマンドフォーマットは次の通りです。

・ldapsearch -LLL -x -D 検索アカウント -W -b 検索先 filter

|  |  |
| --- | --- |
| -LLL | 検索結果を簡易的に表示 |
| -x | 簡易認証で検索 |
| -D 検索アカウント(bindDN) | 検索実行アカウントを指定  （どのアカウントでコマンドを実施するか） |
| -W | パスワードをプロンプト上で入力 |
| -b 検索先(baseDN) | 検索先を指定（どの階層を検索するか） |
| filter | LDAP検索フィルタ |

・ldapmodify -x -W -D 検索アカウント

|  |  |
| --- | --- |
| -x | 簡易認証で検索 |
| -D 検索アカウント(bindDN) | 検索実行アカウントを指定  （どのアカウントでコマンドを実施するか） |
| -W | パスワードをプロンプト上で入力 |
| 標準入力 | LDAP Data Interchange Format |

パスワード、bindDN、baseDNは、Shibboleth IdPからLDAPサーバにアクセスする時のものを使えばよいので、Shibboleth IdPの設定ファイルから取り出します。

※ secrets.propertiesのみ /opt/shibboleth-idp/credentials/ に、それ以外は /opt/shibboleth-idp/conf/ 以下にあります。

|  |
| --- |
| **attribute-resolver.xml内**  　principalCredential="%{idp.attribute.resolver.LDAP.bindDNCredential}"  **secrets.properties内**  　idp.attribute.resolver.LDAP.bindDNCredential =%{idp.authn.LDAP.bindDNCredential:undefined}  　idp.authn.LDAP.bindDNCredential  **=xxxxxxx←パスワード** |
| **attribute-resolver.xml内**  　baseDN="%{idp.attribute.resolver.LDAP.baseDN}"  **ldap.properties内**  　idp.attribute.resolver.LDAP.baseDN = %{idp.authn.LDAP.baseDN:undefined}  　idp.authn.LDAP.baseDN = **o=test\_o,dc=ac,c=JP←baseDN** |
| **attribute-resolver.xml内**  　principal="%{idp.attribute.resolver.LDAP.bindDN}"  **ldap.properties内**  　idp.attribute.resolver.LDAP.bindDN = %{idp.authn.LDAP.bindDN:undefined}  　idp.authn.LDAP.bindDN = **cn=olmgr,o=test\_o,dc=ac,c=JP←bindDN** |

ldapsearch, ldapmodifyコマンドで暗号化されたSeedをldapのbusinessCategory属性に登録します。

|  |
| --- |
| アカウントtest001の属性を確認します。  # ldapsearch -x -b 'o=test\_o,dc=ac,c=JP' -W -LLL -D 'cn=olmgr,o=test\_o,dc=ac,c=JP' uid=test001  Enter LDAP Password: パスワードを入力  dn: uid=test001,ou=Test Unit1,o=test\_o,dc=ac,c=JP  objectClass: eduPerson  objectClass: inetOrgPerson  uid: test001  ou: Test Unit1  ou;lang-ja:: 44OG44K544OIMDAxX+WtpumDqDE=  sn: test001\_sn  sn;lang-ja:: 44OG44K544OIMDAxX3Nu  cn: test001\_cn  userPassword:: dGVzdDAwMQ==  givenName: test001\_givenname  givenName;lang-ja:: 44OG44K544OIMDAxX2dpdmVubmFtZQ==  displayName: test001\_displayname  displayName;lang-ja:: 44OG44K544OIMDAxX2Rpc3BsYXluYW1l  mail: test001\_email@nii.ac.jp  eduPersonAffiliation: member  employeeNumber: 0001  暗号化されたSeedをbusinessCategory属性に追加します。  # ldapmodify -x -W -D 'cn=olmgr,o=test\_o,dc=ac,c=JP'  Enter LDAP Password: パスワードを入力  dn: uid=test001,ou=Test Unit1,o=test\_o,dc=ac,c=JP  changetype: modify  add: businessCategory  businessCategory: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*←暗号化されたSeed  CTRL+d←ctrlキーを押しながらdを押す  modifying entry "uid=test001,ou=Test Unit1,o=test\_o,dc=ac,c=JP"  businessCetegoryに登録されたか確認します。  # ldapsearch -x -b 'o=test\_o,dc=ac,c=JP' -W -LLL -D 'cn=olmgr,o=test\_o,dc=ac,c=JP' uid=test001  Enter LDAP Password: パスワードを入力  dn: uid=test001,ou=Test Unit1,o=test\_o,dc=ac,c=JP  objectClass: eduPerson  objectClass: inetOrgPerson  uid: test001  ou: Test Unit1  ou;lang-ja:: 44OG44K544OIMDAxX+WtpumDqDE=  sn: test001\_sn  sn;lang-ja:: 44OG44K544OIMDAxX3Nu  cn: test001\_cn  userPassword:: dGVzdDAwMQ==  givenName: test001\_givenname  givenName;lang-ja:: 44OG44K544OIMDAxX2dpdmVubmFtZQ==  displayName: test001\_displayname  displayName;lang-ja:: 44OG44K544OIMDAxX2Rpc3BsYXluYW1l  mail: test001\_email@nii.ac.jp  eduPersonAffiliation: member  employeeNumber: 0001  businessCategory: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*←暗号化されたSeed  ↑businessCategoryに登録されていることを確認できます。 |

**※ldapmodifyは、ldifファイルにコマンドを設定して実行する方法もあります。**

* /root/mod\_test001.ldifとして、Seed値設定用のldifファイルを作成します。

|  |
| --- |
| dn: uid=test001,ou=Test Unit1,o=test\_o,dc=ac,c=JP  changetype: modify  replace: businessCategory  businessCategory: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  暗号化されたSeed↑ |

* 以下コマンドを実行して、businessCategoryにSeed値を設定します。  
  LDAP Passwordの値は、/opt/shibboleth-idp/credentials/secrets.propertiesのidp.authn.LDAP.bindDNCredentialに記載されています。

|  |
| --- |
| # ldapadd -W -D cn=olmgr,o=test\_o,dc=ac,c=JP -f /root/mod\_test001.ldif Enter LDAP Password: パスワードを入力してEnter  modifying entry "uid=test001,ou=Test Unit1,o=test\_o,dc=ac,c=JP" |

# 

# **3. 手順書**

以下は、Shibboleth公式のTOTPプラグインに関するマニュアルのURLです。詳細にご興味がある方はご参照ください。

参考:<https://shibboleth.atlassian.net/wiki/spaces/IDPPLUGINS/pages/1376878877/TOTP>

# **4. 動作確認**

## **4.1. TOTP対応アプリの設定**

① ブラウザで「[2.2. TOTPシード作成](#_heading=h.8khrjl3oigla)」で保存した**QR CodeのURL**を開き、QR Codeを表示します。

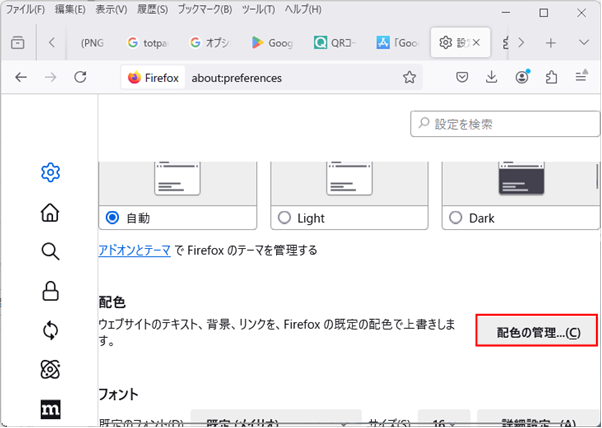


#### **図　シード用QRコードの表示画面**

ここでブラウザの背景色が黒で、QRコードを読み取りにくい時は、ブラウザの設定で背景色を白くしてみてください。

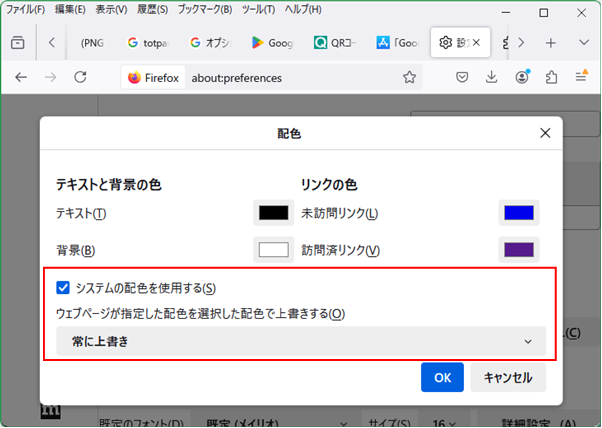
* Firefoxで背景色を変える方法

設定画面の「配色の管理」をクリックしてください。



#### **図　Firefoxの設定画面**

「システムの配色を使用する」「常に上書き」を選び、OKをクリックしてください。



#### **図　FireFoxでシステムの背景色（白）で上書する**

* MS Edgeで背景色を変える方法

「設定」画面で「ページの色」で「白」を選択し、「ハイコントラストテーマがオンの場合にのみページのカラーを適用する」をオフにしてください。



#### **図　MS Edgeの設定画面**

② TOTP対応アプリ（例: Google Authenticator）を起動します。

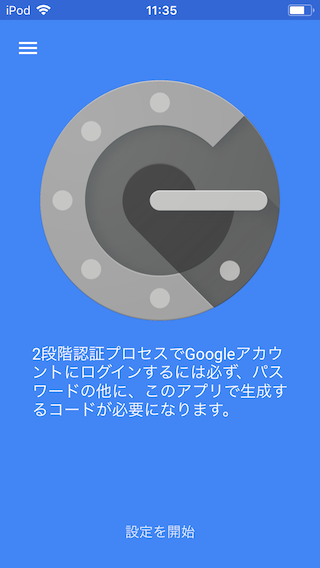
※Google Authenticatorは、Android用はGooglePlayで、iPhone用はAppStoreで、「Google Authenticator」をキーとして検索してダウンロードできます。ただし色々なものが検索結果に表示されるので、Google社が提供するものを選んでください。あるいは以下のQRコードをご利用ください。

※次のQRコードでもGoogle Authenticatorアプリをダウンロードできます。

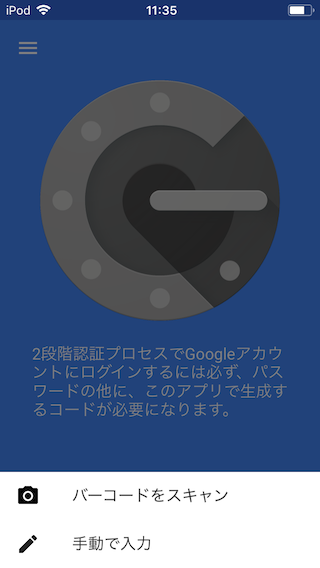
**Android用**　　　　　　　　　　　　　　　　　**iPhone用**

③ 「設定を開始」をタップします。

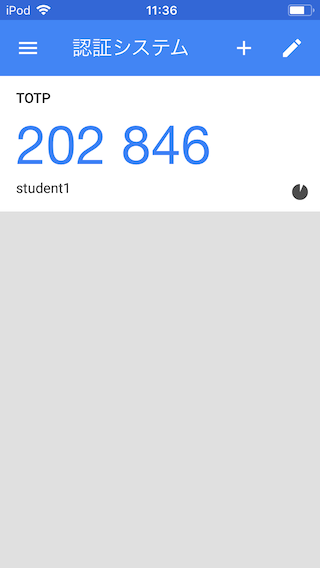


④ 「バーコードをスキャン」をタップします。



⑤ 表示されているQRコードを読み取ります。

⑥ 登録後に数字6桁のワンタイムパスワードが表示されます。



## 

## **4.2. ログイン**

① 接続確認用SPにアクセスします。

テストフェデレーションの場合:

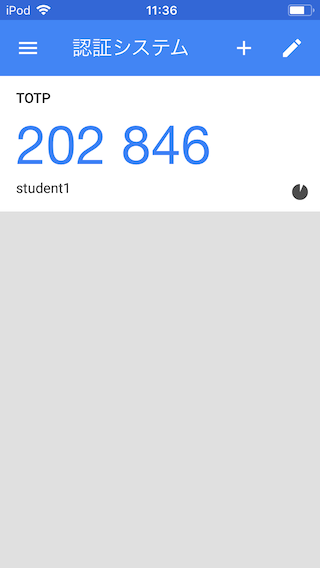
<https://test-sp1.gakunin.nii.ac.jp>

② ログインボタンをクリックします。

③ DSの設定を行っている場合、所属機関の選択画面が表示されるので、各自が使用するIdP（今回、TOTP認証を設定したIdPサーバ）を選択します。

④ 第一画面は通常と変わりませんので、通常のUsername/Passwordを入力して次に進みます。

⑤ "Token code"というワンタイムパスワードを入力する画面になりますので、TOTP対応アプリ/デバイスに表示されている数字6桁のワンタイムパスワードを入力して次に進みます。



⑥ 送信属性同意画面が表示されます。そのまま設定値を送信します。

⑦ 認証後、正しく属性受信の確認ページが表示される事を確認してください。