2024年11月8日

# Shibboleht IdPへのWebAuthnプラグイン導入例

国立情報学研究所

教育研修事業

[**Shibboleht IdPへのWebAuthnプラグイン導入例**](#_heading=h.gjdgxs) **1**

[**1．はじめに**](#_heading=h.30j0zll) **1**

[**2．前提条件**](#_heading=h.1fob9te) **2**

[**3. 検証環境**](#_heading=h.3znysh7) **2**

[**4．システム構成**](#_heading=h.2et92p0) **3**

[**5．動作概要**](#_heading=h.tyjcwt) **3**

[**6．Shibboleth IdPの変更**](#_heading=h.3dy6vkm) **4**

[**7．Shibboleth SPの変更**](#_heading=h.1t3h5sf) **8**

[**8．動作確認**](#_heading=h.4d34og8) **9**

[資格情報登録フロー](#_heading=h.2s8eyo1) 9

[ログインフロー](#_heading=h.17dp8vu) 14

# 1．はじめに

このドキュメントではShibboleth IdPバージョン5へのWebAuthnプラグインの導入の例を紹介しています。

このドキュメントで構築する検証環境の動作確認にはFIDO2対応認証デバイスが必要です。検証環境ではFIDO2対応認証デバイスとしてWindows 11でパスキーを使って説明しています。

お使いの端末でパスキーを試したい方は次のデモページを参照してください。

* <https://www.passkeys.io/>

|  |
| --- |
| ※FIDO2（ファイド2）は、パスワードレス認証を実現する技術規格で、業界団体であるFIDOアライアンスが2018年に発表しました。従来のパスワード認証に代わる認証手段として期待されており、セキュリティを高めながらユーザーの利便性を向上させることが期待できます。﻿  FIDO2の主な特徴は次のとおりです。﻿   * 専用のソフトウェアやハードウェアを必要とせず、指紋認証や顔認証、虹彩認証などの生体情報をオンライン上でやり取りできる﻿ * 公開鍵暗号方式を利用しており、資格情報（指紋情報、顔情報）を認証サーバーと共有しないため、フィッシングやキーロガー、リプレイ攻撃、中間者攻撃によるリスクが軽減される﻿ * モバイルとデスクトップの両方の環境でオンラインサービスを簡単に認証できる﻿   FIDO2の認証処理は、FIDOサーバーと呼ばれる認証専用サーバーで行われます。Webサービスと連携し、クライアントとFIDOサーバーとの間で認証指示（チャレンジコード）のやり取りを行うことで認証処理を実行します。﻿ |

# 2．前提条件

このドキュメントで説明する検証環境はすでにShibboleth IdP と SP が構成され、SAML SSO の準備ができていることを前提としています。IdP はSP の認証要求ごとにユーザーを認証し、ユーザーの属性を適切に SP に返すことができる必要があります。

また検証環境はGakuNin が提供する次のドキュメントに基づいてインストールしていることを前提としています。異なる方法でインストールした場合は、システムの要件に応じて以降の手順を変更してください。

* <https://meatwiki.nii.ac.jp/confluence/pages/viewpage.action?pageId=157188958>（日本語）

# 3. 検証環境

検証環境で使用しているソフトウェアのバージョンは次のとおりです。

* Rocky Linux 9.4
* Apache httpd 2.4
* OpenJDK 17.0.12
* Jetty 11.0.20
* Shibboleth IdP 5.1.2
  + WebAuthnプラグイン 0.0.2
* Shibboleth SP 3.4.1

# 4．システム構成

検証環境のシステム構成要素は次のとおりです。

* Shibboleth IdP［FQDN: ex-idp-test??.gakunin.nii.ac.jp］
  + ユーザー向けにWebAuthnの資格情報を登録・削除するためのWebインタフェースを提供します。
  + Shibboleth SPにWebAuthnによる認証フローを提供します。
* Shibboleth SP［FQDN: ex-sp-test??.gakunin.nii.ac.jp］
  + WebAuthnによる認証を表示要件とするWebサービスです。検証環境では受信した属性情報一覧を表示するページです。

# 5．動作概要

検証環境ではシステム全体が次のように動作するように設定します。

* WebAuthn資格情報登録フロー
  + IdPではユーザー名とパスワードの入力により認証し、デフォルトのIdPストレージにWebAuthnの資格情報（ユーザー名とFIDO2認証器で生成された公開鍵など）を登録します。
* SP認証フロー
  + ユーザーはSP経由でのIdPでの認証時にユーザー名とパスワードの入力が不要となり、WindowsやmacOSのパスキーによるジェスチャー（PINの入力やTouch ID等の指紋認証、スマートフォンのタップなど）や、FIDO2に対応した認証デバイスにより認証が完了します。

WebAuthnプラグインには、他にも次のような制約調整項目や認証フローへの対応など、様々な設定項目があります。

* ユーザーIDまたはパスワードの入力を必須とする
* MFAの一要素として設定する

詳しくは次のドキュメントを参照してください。

* <https://shibboleth.atlassian.net/wiki/spaces/IDPPLUGINS/pages/3395321933/WebAuthnAuthnConfiguration>

# 6．Shibboleth IdPの変更

IdPにWebAuthnプラグインを導入します。

1. plugin.shコマンドでWebAuthnプラグインをインストールします。

|  |
| --- |
| # /opt/shibboleth-idp/bin/plugin.sh --noCheck -i https://shibboleth.net/downloads/prerelease/idp-plugin-webauthn-0.0.2.tar.gz  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/admin/admin.properties  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/authn/authn.properties  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/c14n/subject-c14n.properties  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/services.properties  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/saml-nameid.properties  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/ldap.properties  INFO - Downloading from HTTPResource [https://shibboleth.net/downloads/prerelease/idp-plugin-webauthn-0.0.2.tar.gz]  ...................  INFO - Downloading from HTTPResource [https://shibboleth.net/downloads/prerelease/idp-plugin-webauthn-0.0.2.tar.gz.asc]  INFO - Plugin net.shibboleth.idp.plugin.authn.webauthn: Trust store folder does not exist, creating  INFO - Plugin net.shibboleth.idp.plugin.authn.webauthn: Trust store does not exist, creating  INFO - TrustStore does not contain signature 0x7D27E610B8A3DC52  Accept this key:  Signature: 0x7D27E610B8A3DC52  FingerPrint: B5B5DD332142AD657E8D87AC7D27E610B8A3DC52  Username: Philip David Smart <philip.smart@jisc.ac.uk>  [yN] y ← ‘y’を入力して[Enter]キーを押下。  INFO - Installing Plugin 'net.shibboleth.idp.plugin.authn.webauthn' version 0.0.2  INFO - Rebuilding /opt/shibboleth-idp/war/idp.war, Version 5.1.2  INFO - Initial populate from /opt/shibboleth-idp/dist/webapp to /opt/shibboleth-idp/webpapp.tmp  INFO - Overlay from /opt/shibboleth-idp/dist/plugin-webapp to /opt/shibboleth-idp/webpapp.tmp  INFO - Overlay from /opt/shibboleth-idp/edit-webapp to /opt/shibboleth-idp/webpapp.tmp  INFO - Creating war file /opt/shibboleth-idp/war/idp.war  INFO - Module file changes as a result of this install  INFO - conf/authn/webauthn.properties created  INFO - views/webauthn/webauthn-register-username.vm created  INFO - conf/authn/webauthn-config.xml created  INFO - views/webauthn/webauthn-register.vm created  INFO - edit-webapp/js/webauthn/webauthn-json.browser-ponyfill.js created  INFO - views/webauthn/webauthn-authn-username.vm created  INFO - edit-webapp/css/webauthn.css created  INFO - views/webauthn/webauthn-authn.vm created  INFO - edit-webapp/js/webauthn/webauthn-support.js created  Plugin: net.shibboleth.idp.plugin.authn.webauthn Current Version: 0.0.2  # /opt/shibboleth-idp/bin/module.sh -l  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/authn/authn.properties  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/authn/webauthn.properties  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/services.properties  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/ldap.properties  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/saml-nameid.properties  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/admin/admin.properties  INFO - Including auto-located properties in /opt/shibboleth-idp/bin/../conf/c14n/subject-c14n.properties  Module: idp.authn.WebAuthn [ENABLED] ← WebAuthnモジュールが有効になっていることを確認。  Module: idp.Core [ENABLED]  Module: idp.CommandLine [ENABLED]  Module: idp.EditWebApp [ENABLED]  Module: idp.UserPrefs [DISABLED]  Module: idp.authn.IPAddress [DISABLED]  Module: idp.authn.MFA [DISABLED]  Module: idp.authn.Password [ENABLED]  Module: idp.authn.SPNEGO [DISABLED]  Module: idp.authn.X509 [DISABLED]  Module: idp.authn.Demo [DISABLED]  Module: idp.admin.Hello [ENABLED]  Module: idp.admin.UnlockKeys [DISABLED]  Module: idp.intercept.Consent [DISABLED]  Module: idp.intercept.ContextCheck [DISABLED]  Module: idp.intercept.ExpiringPassword [DISABLED]  Module: idp.intercept.Impersonate [DISABLED]  Module: idp.intercept.Warning [DISABLED]  Module: idp.profile.CAS [DISABLED] |

1. 1.で導入したプラグイン（バージョン0.0.2）には、後述のステップで必要なクラスが不足しているため別途配布のjarファイル（idp-plugin-webauthn-impl-0.0.2.1.jar　MD5チェックサム値：5995ab71bdbd72726c8b0095c094141d）に置き換えてidp.warファイルをリビルドします。

|  |
| --- |
| # md5sum /root/PKG/idp-plugin-webauthn-impl-0.0.2.1.jar  5995ab71bdbd72726c8b0095c094141d /root/idp-plugin-webauthn-impl-0.0.2.1.jar  # rm /opt/shibboleth-idp/dist/plugin-webapp/WEB-INF/lib/idp-plugin-webauthn-impl-0.0.2.jar  # cp /root/PKG/idp-plugin-webauthn-impl-0.0.2.1.jar /opt/shibboleth-idp/dist/plugin-webapp/WEB-INF/lib/  # /opt/shibboleth-idp/bin/build.sh  IINFO - net.shibboleth.idp.installer.impl.IdPBuildArguments@3c5a99da  INFO - Rebuilding /opt/shibboleth-idp/bin/../war/idp.war, Version 5.1.2  INFO - Initial populate from /opt/shibboleth-idp/bin/../dist/webapp to /opt/shibboleth-idp/bin/../webpapp.tmp  INFO - Overlay from /opt/shibboleth-idp/bin/../dist/plugin-webapp to /opt/shibboleth-idp/bin/../webpapp.tmp  INFO - Overlay from /opt/shibboleth-idp/bin/../edit-webapp to /opt/shibboleth-idp/bin/../webpapp.tmp  INFO - Creating war file /opt/shibboleth-idp/bin/../war/idp.war |

1. /opt/shibboleth-idp/conf/authn/authn.propertiesファイルを次のように変更し、WebAuthn認証方式を有効にします。

|  |
| --- |
| idp.authn.flows = Password|WebAuthn |

1. /opt/shibboleth-idp/conf/authn/webauthn.propertiesファイルを次のように変更し、SAMLリクエストで特定のauthnContextClassRefが指定されたときのみWebAuthn認証方式が選択されるように変更します。（この設定はWebAuthn認証フローを単純に説明する目的の仮の設定のため、運用では使用しないでください）

|  |
| --- |
| idp.authn.webauthn.relyingPartyId = ex-idp-test??.gakunin.nii.ac.jp  idp.authn.webauthn.relyingPartyName = Shibboleth  idp.authn.webauthn.supportedPrincipals = \  saml2/urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered  idp.authn.webauthn.admin.registration.accessPolicy = AccessByCurrentUser |

1. /opt/shibboleth-idp/conf/access-control.xml ファイルを次のように変更し、資格情報登録フローにアクセス制限を追加します。この例では（パスワード認証により）認証されたユーザのみ資格情報の登録が可能です。  
   <util:map id="shibboleth.AccessControlPolicies">～</util:map>の間に、次の<entry key="AccessByCurrentUser">～</entry>を挿入してください。

|  |
| --- |
| <util:map id="shibboleth.AccessControlPolicies">  …  <entry key="AccessByCurrentUser">  <bean parent="shibboleth.PredicateAccessControl">  <constructor-arg>  <bean id="AccessByCurrentUserPredicate"  class="net.shibboleth.idp.plugin.authn.webauthn.admin.impl.AllowCurrentUserAccessPredicate">  </bean>  </constructor-arg>  </bean>  </entry>  …  </util:map> |

本手順は公式ページ（<https://shibboleth.atlassian.net/wiki/spaces/IDPPLUGINS/pages/3395321933/WebAuthnAuthnConfiguration>）で紹介されていますが、プラグインバージョン0.0.2で導入されるjarファイルには上記設定で参照しているAllowCurrentUserAccessPredicateクラスが存在しないため資格情報登録フローが動作しません。このため、2.でAllowCurrentUserAccessPredicateクラスを含むjarファイルに置き換えています。

なおプラグインバージョン0.0.3にはAllowCurrentUserAccessPredicateクラスが含まれますが、2024年8月時点ではIdP起動時にエラーが発生するため採用を見送っています。

1. Jettyを再起動します。

|  |
| --- |
| # systemctl stop jetty  # systemctl start jetty |

# 7．Shibboleth SPの変更

SPの認証要求をWebAuthn認証に変更します。

1. /etc/httpd/conf.d/shib.conf を次のように変更してください。

|  |
| --- |
| <Location /secure>  AuthType shibboleth  ShibRequestSetting requireSession 1  ShibRequestSetting authnContextClassRef urn:oasis:names:tc:SAML:2.0:ac:classes:MobileOneFactorUnregistered  require shib-session  </Location> |

1. httpdを再起動します。

|  |
| --- |
| # systemctl stop httpd  # systemctl start httpd |

# 8．動作確認

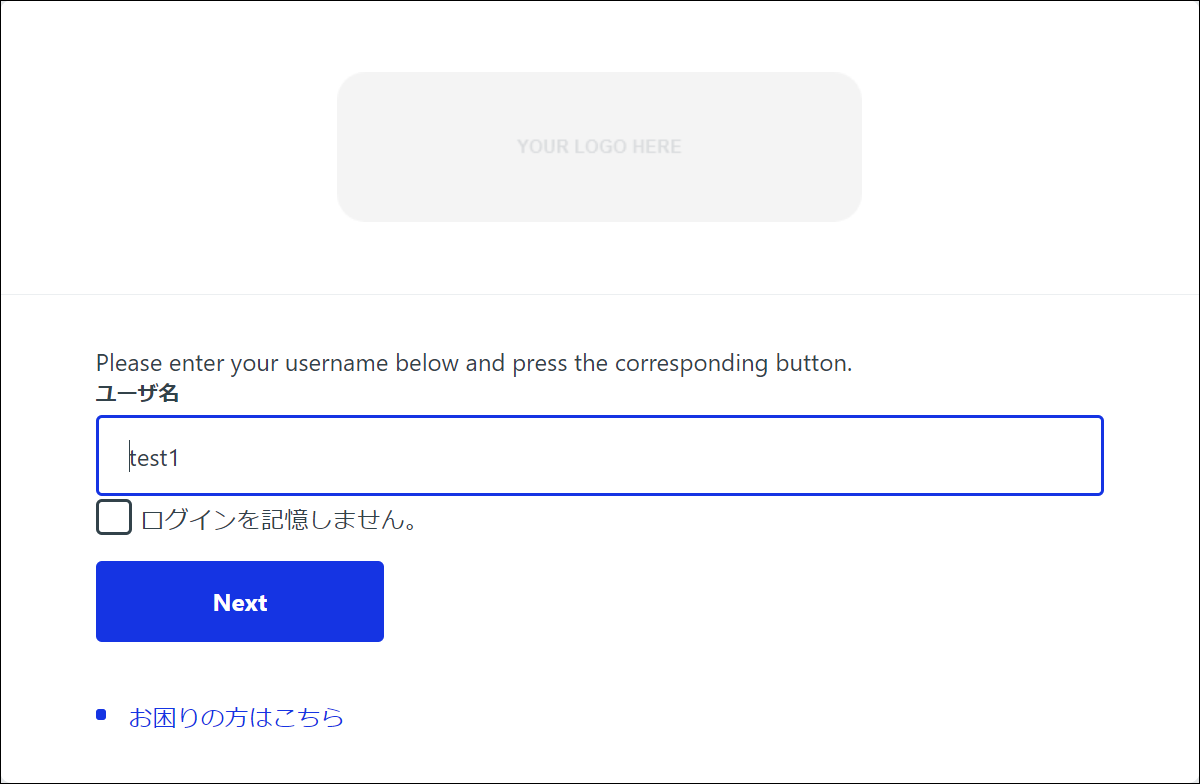
このドキュメントではWindows 11およびChromeブラウザで動作確認しています。

（macOSでも表示される画面や名称に違いはありますが同様の手順で確認可能です）

### 資格情報登録フロー

1. PCのブラウザでIdPの資格情報登録ページを表示します。

|  |
| --- |
| https://ex-idp-test??.gakunin.nii.ac.jp/idp/profile/admin/webauthn-registration |

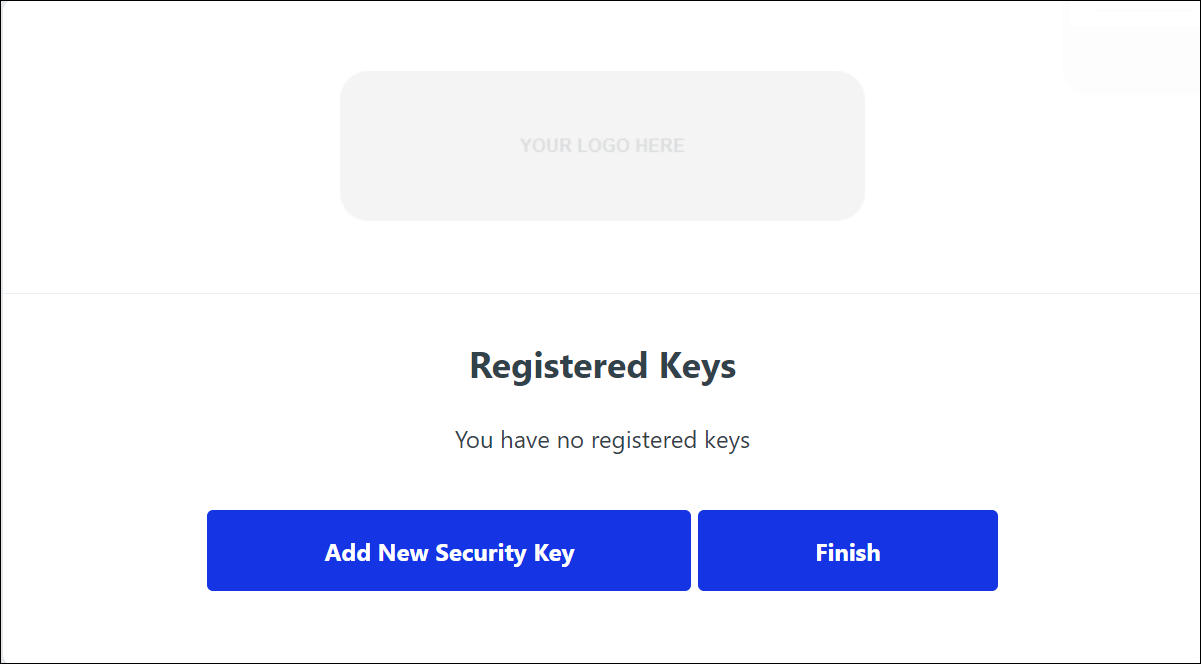


［ユーザ名］を入力して［Next］ボタンを押下します。

1. 続いて表示される認証画面でもう一度［ユーザ名］と、［パスワード］を入力して［Login］ボタンを押下します。

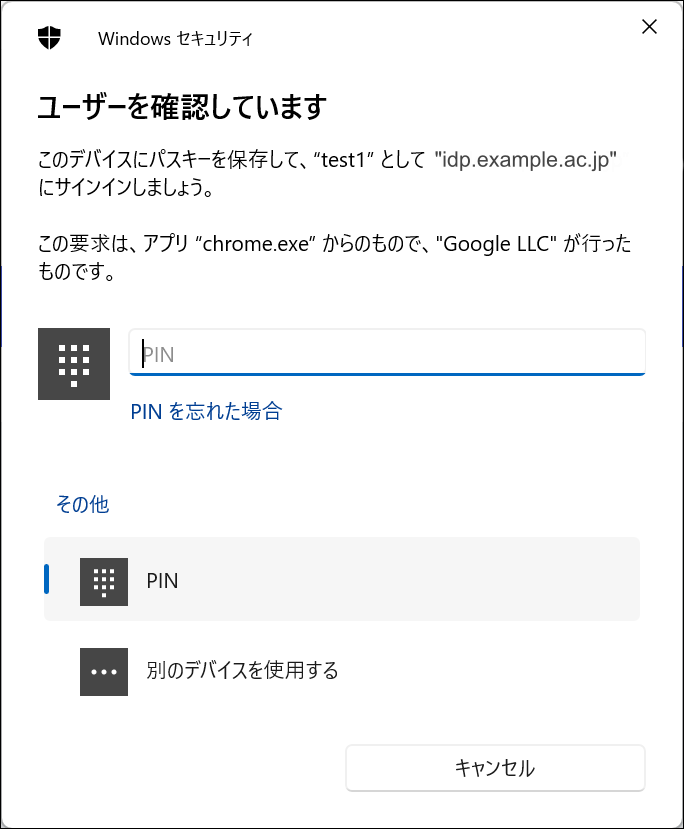


1. IdPのパスワード認証を通過すると、セキュリティキーの登録画面が表示されます。



［Add New Security Key］ボタンを押下して、新しいセキュリティキーの登録に進みます。

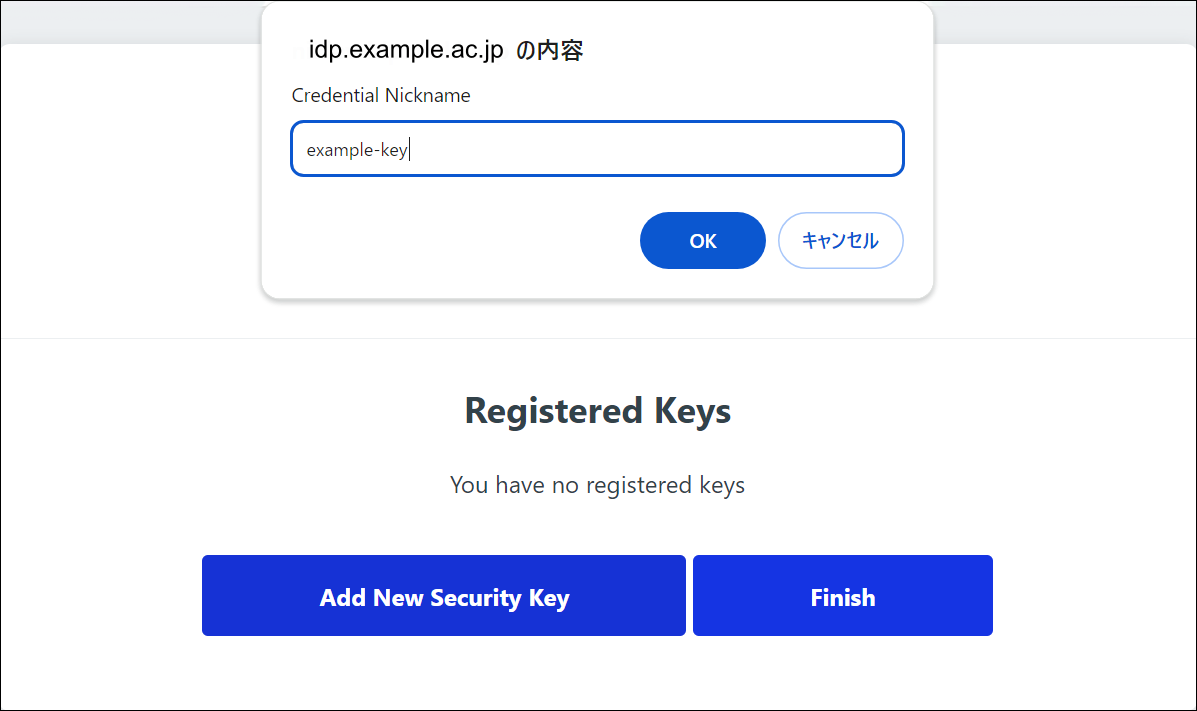
1. Windows Helloの認証画面が表示されます。（この例ではPINコードを入力しています）



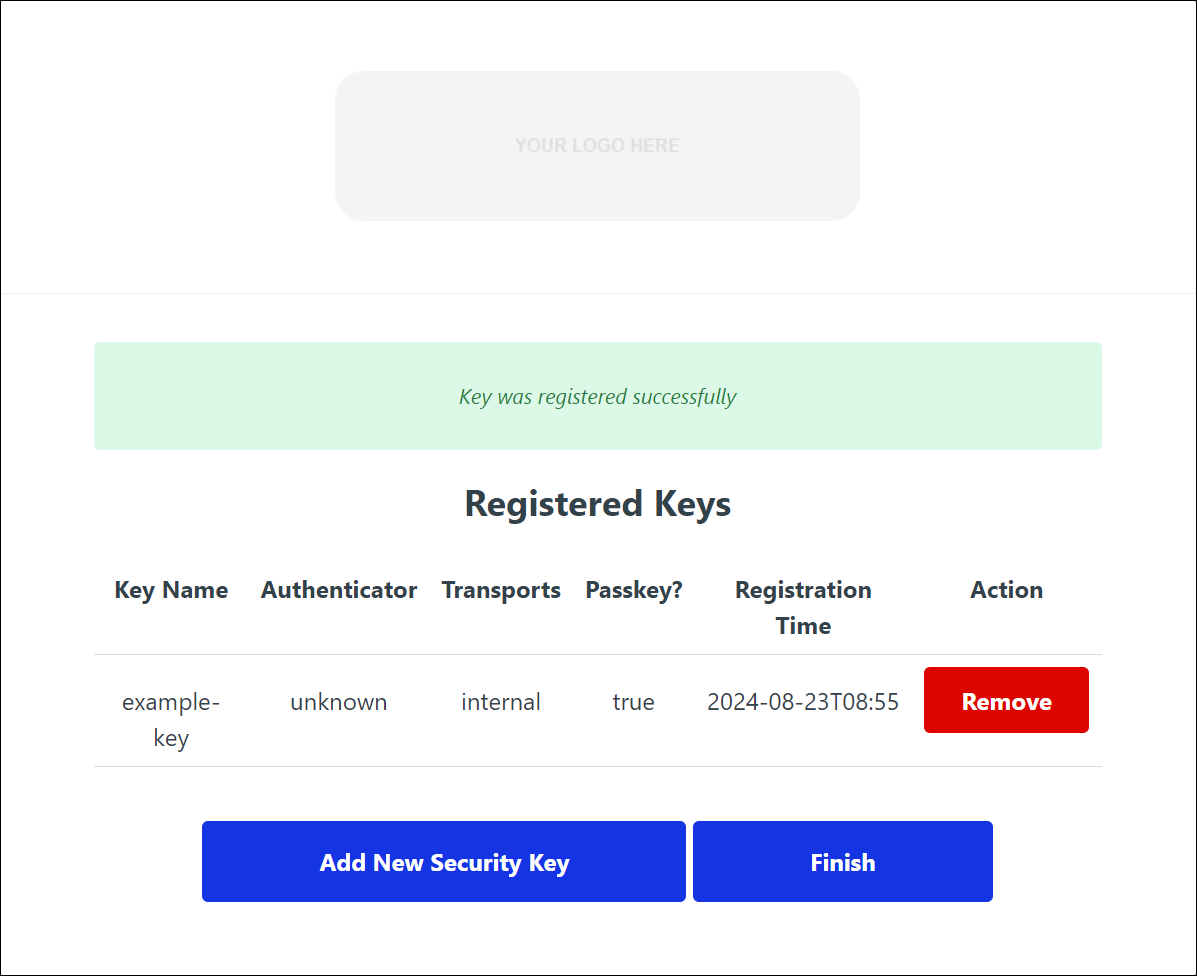
1. Windows Helloの認証を通過すると、IdPの情報がパスキーとして登録されます。



［OK］ボタンを押下すると、IdPのストレージに保存されるセキュリティキー名の入力画面ポップアップが表示されるので、適当な名前を入力して［OK］ボタンを押下します。



1. IdPにパスキーの情報がセキュリティキーとして登録されます



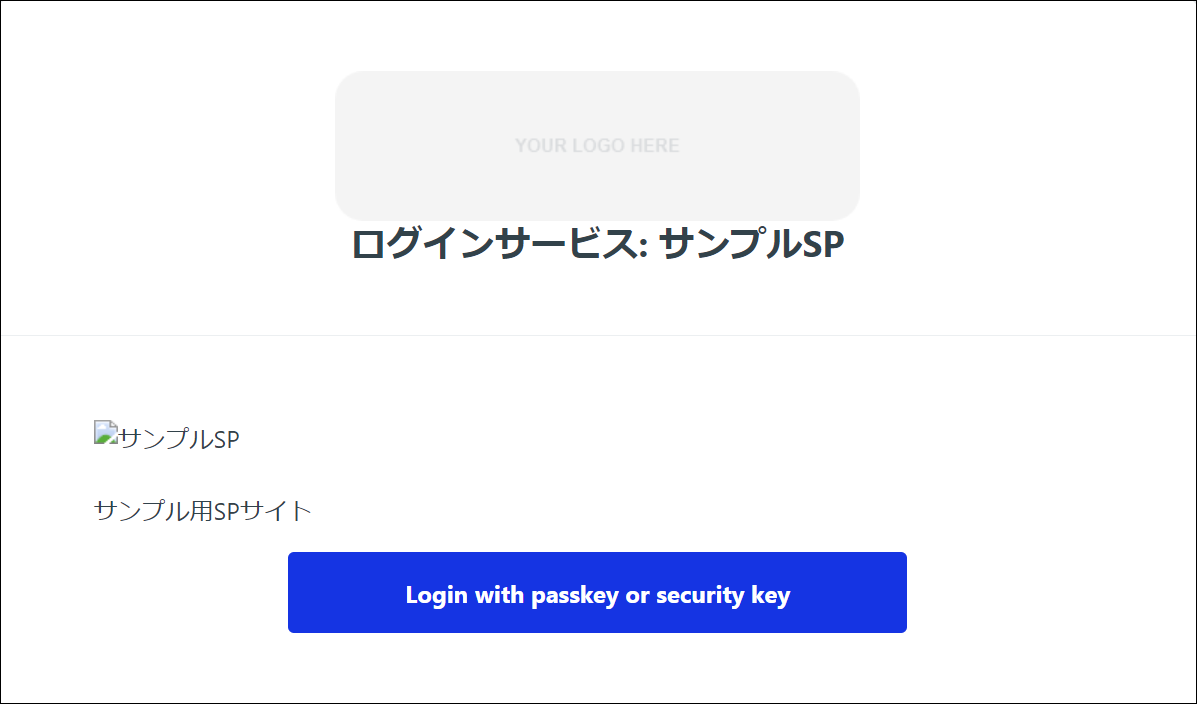
［Finish］ボタンを押下して登録を終了します。

なおこの画面ではIdPに保存されているセキュリティキーの情報の削除が可能です。

### ログインフロー

1. PCのブラウザで各自が使用するSPの接続確認用ページにアクセスします。

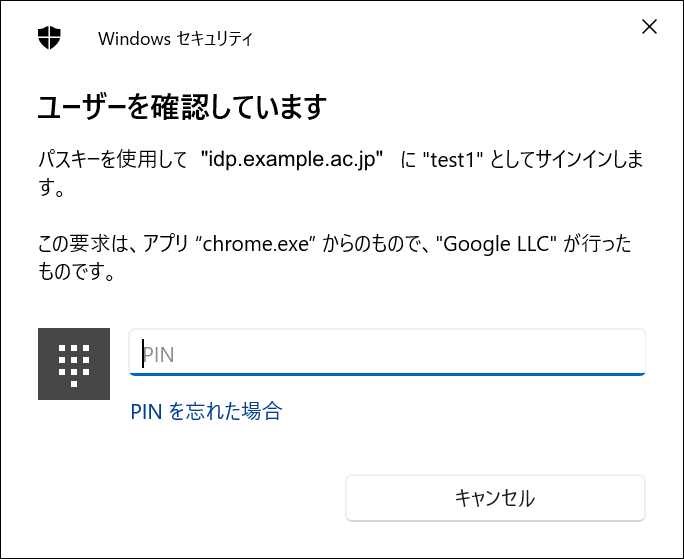
|  |
| --- |
| 例）1番を割り振られた場合  　https://ex-sp-test01.gakunin.nii.ac.jp/ |

1. ログインボタンをクリックします。
2. 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を選択して、IdPのログイン画面を表示します。  
   
3. ［Login with passkey or security key］ボタンを押下します。



ポップアップでパスキーの使用可否について聞かれますので［続行］ボタンを押下します。

1. Windows Helloの認証画面が表示されます。（この例ではPINコードを入力しています）



1. Windows Helloの認証を通過するとSPの画面が表示されます。  
   