SafeNet Authentication Service
Integration Guide

Using SAS as an Identity Provider for Microsoft Azure Access Control Service
All information herein is either public information or is the property of and owned solely by Gemalto NV. and/or its subsidiaries who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Gemalto’s information.

This document can be used for informational, non-commercial, internal and personal use only provided that:

- The copyright notice below, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- This document shall not be posted on any network computer or broadcast in any media and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided “AS IS” without any warranty of any kind. Unless otherwise expressly agreed in writing, Gemalto makes no warranty as to the value or accuracy of information contained herein.

The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, Gemalto reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

Gemalto hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Gemalto be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Gemalto does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms’ resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Gemalto be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Gemalto products. Gemalto disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service or loss of privacy.

© 2015 Gemalto. All rights reserved. Gemalto and the Gemalto logo are trademarks and service marks of Gemalto N.V. and/or its subsidiaries and are registered in certain countries. All other trademarks and service marks, whether registered or not in specific countries, are the property of their respective owners.

Document Part Number: 007-013261-001, Rev. A
Release Date: October 2015
Contents

Third-Party Software Acknowledgement ................................................................................................................. 4
Description ........................................................................................................................................................................ 4
Applicability ....................................................................................................................................................................... 4
Environment ........................................................................................................................................................................... 5
Audience .............................................................................................................................................................................. 5
SAML Authentication using SAS Cloud ......................................................................................................................... Error! Bookmark not defined.
SAML Authentication using SAS-SPE and SAS-PCE ................................................................................................ Error! Bookmark not defined.
SAML Authentication Flow using SAS ....................................................................................................................... 5
Configuring a SAS Auth Node and Encryption Key ................................................................................................. 6
Configuring Microsoft Azure Access Control Service and AD FS ............................................................................ 8
    Creating a Web Application to be Secured by ACS .................................................................................................. 8
    Integrating the Web Application with ACS ............................................................................................................. 9
    Testing the Integration with ACS .......................................................................................................................... 14
    Viewing the Application in the ACS Management Portal .................................................................................. 15
    Viewing Claims Sent by ACS .................................................................................................................................. 18
    Adding AD FS as an Identity Provider of ACS ...................................................................................................... 20
    Configuring a Trust Relationship Between AD FS and ACS ............................................................................. 23
    Configuring the ACS Relying Party Application's Rules ..................................................................................... 25
    Configuring the AD FS Agent .................................................................................................................................. 26
    Configuring the AD FS Authentication Policy ..................................................................................................... 28
Configuring SafeNet Authentication Service ........................................................................................................ 29
    Synchronizing Users Stores to SAS ........................................................................................................................ 29
    Assigning an Authenticator in SAS ......................................................................................................................... 30
Running the Solution ..................................................................................................................................................... 31
Support Contacts .......................................................................................................................................................... 33
Third-Party Software Acknowledgement

This document is intended to help users of SafeNet products when working with third-party software, such as Microsoft Azure Access Control Service (ACS).

Material from third-party software is being used solely for the purpose of making instructions clear. Screen images and content obtained from third-party software will be acknowledged as such.

Description

SafeNet Authentication Service delivers a fully automated, versatile, and strong authentication-as-a-service solution.

With no infrastructure required, SafeNet Authentication Service provides smooth management processes and highly flexible security policies, token choice, and integration APIs.

Access Control Service or Windows Azure Access Control Service is a Microsoft-owned cloud-based service that provides an easy way of authenticating and authorizing users to gain access to web applications and services, while allowing the features of authentication and authorization to be factored out of the application code.

This document describes how to:

- Deploy multi-factor authentication (MFA) options in Microsoft Azure Access Control Service using SafeNet one-time password (OTP) authenticators managed by SafeNet Authentication Service.
- Configure SAML authentication in Microsoft Azure Access Control Service using SafeNet Authentication Service as an identity provider.

It is assumed that the Microsoft Azure Access Control Service environment is already configured and working with static passwords prior to implementing multi-factor authentication using SafeNet Authentication Service.

Microsoft Azure Access Control Service can be configured to support multi-factor authentication in several modes. The SAML authentication will be used for the purpose of working with SafeNet Authentication Service.

Applicability

The information in this document applies to:

- **SafeNet Authentication Service (SAS)**—SafeNet's cloud-based authentication service
- **SafeNet Authentication Service – Service Provider Edition (SAS-SPE)**—A server version that is used by service providers to deploy instances of SafeNet Authentication Service
- **SafeNet Authentication Service – Private Cloud Edition (SAS-PCE)**—A server version that is used to deploy the solution on-premises in the organization
Environment
The integration environment that was used in this document is based on the following software versions:

- **SafeNet Authentication Service**—Version 3.3
- **Microsoft Azure Access Control Service**—included in the MS Azure portal
- **AD FS**—in Windows Server 2012 R2

Audience
This document is targeted to system administrators who are familiar with Microsoft Azure Access Control Service, and are interested in adding multi-factor authentication capabilities using SafeNet Authentication Service.

SAS Authentication API Authentication using SAS Cloud
SAS Cloud provides a service for SAS Authentication API that is already implemented in the SAS Cloud environment and can be used using the Gemalto SafeNet SAS ADFS Agent.

SAS Authentication API using SAS-SPE and SAS-PCE
In addition to the pure cloud-based offering, SafeNet Authentication Service comes with two on-premises versions:

- **SafeNet Authentication Service – Service Provider Edition (SPE)**—An on-premises version of SafeNet Authentication Service targeted at service providers interested in hosting SAS in their data center.
- **SafeNet Authentication Service – Private Cloud Edition (PCE)**—An on-premises version of SafeNet Authentication Service targeted at organizations interested in hosting SAS in their private cloud environment.

For both on-premises versions, SAS can be integrated with AD FS infrastructure, which uses a special on-premises agent called Gemalto SafeNet SAS ADFS Agent.
SAS Authentication API Flow using SAS

AD FS provides extensible multi-factor authentication through the concept of “additional authentication providers” that are invoked during secondary authentication. External providers can be registered in AD FS.

Once a provider is registered with AD FS, it is invoked from the AD FS authentication code via specific interfaces and methods that the provider implements and that AD FS calls. Because it provides a bridge between AD FS and an external authentication provider, the external authentication provider is also called an AD FS MFA “adapter”.

Gemalto SafeNet SAS ADFS Agent is an AD FS MFA adapter that provides users a way to authenticate through AD FS using SAS as a secondary authenticator.

The image below describes the dataflow of a multi-factor authentication transaction for Microsoft Azure Access Control Service.

1. A user attempts Sign in to Microsoft Azure Access Control Service. The user is redirected AD FS proxy server (WAP), then after successful authentication, is forwarded to SafeNet Authentication Service (SAS) for a secondary authentication (AD FS multi-factor authentication).
2. The user uses his SAS token for authenticating. SAS collects and evaluates the user’s credentials.
3. The SAS authentication reply is sent back to AD FS which returns a response to Microsoft Azure Access Control Service, accepting or rejecting the user’s authentication request.
4. The user is granted or denied access to Microsoft Azure Access Control Service.

Configuring a SAS Auth Node and Encryption Key

If the SAS server is not installed on the same machine as AD and AD FS, perform the following steps:

1. Log in to the SAS console with an Operator account.
2. Click VIRTUAL SERVERS > COMMS > Authentication Processing.
3. Click the Authentication Agent Settings link, and then click Download to download the encryption key file. This file will be needed in step 4 of “Configuring the AD FS Agent” on page 26.
Click VIRTUAL SERVERS > COMMS > Auth Nodes.

5. Click the Auth Nodes link.

6. Click Add.

7. In the Auth Nodes section, complete the following fields, and then click Save:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Description</td>
<td>Enter a description for this node (for example, DC).</td>
</tr>
<tr>
<td>Host Name</td>
<td>Enter the name of the host that will authenticate with SAS.</td>
</tr>
<tr>
<td>Low IP Address In Range</td>
<td>Enter the lowest IP address that will authenticate with SAS.</td>
</tr>
<tr>
<td>High IP Address In Range</td>
<td>Enter the highest IP address that will authenticate with SAS. (The low and high IP addresses may be the same, since the node is referencing a single machine.)</td>
</tr>
<tr>
<td>Exclude from PIN change requests</td>
<td>Do not select this option.</td>
</tr>
</tbody>
</table>
Configuring Microsoft Azure Access Control Service and AD FS

Configuring Microsoft Azure Access Control Service (ACS) and AD FS requires the following:

- Creating a Web Application to be Secured by ACS, page 8
- Integrating the Web Application with ACS, page 9
- Testing the Integration with ACS, page 14
- Viewing the Application in the ACS Management Portal, page 15
- Viewing Claims Sent by ACS, page 18
- Adding AD FS as an Identity Provider of ACS, page 20
- Configuring a Trust Relationship Between AD FS and ACS, page 23
- Configuring the ACS Relying Party Application’s Rules, page 25
- Configuring the AD FS Agent, page 26
- Configuring the AD FS Authentication Policy, page 28

Creating a Web Application to be Secured by ACS

Any web application may be secured by ACS. However, in this guide, we will use an application created by MS Visual Studio 2012. If you already have a web application, you may skip this section.

2. Select the Visual C#/Web template, and then select ASP.NET MVC 4 Web Application.
3. In the Name field, enter MvcACS, and then click OK.
4. On the next window, select Internet Application, and then click OK.
5. Edit the Views\Shared_LoginPartial.cshtml file and replace the content with the following code:

```html
@if (Request.IsAuthenticated) {
    <text>
        Hello, @Html.ActionLink(User.Identity.Name, "Manage", "Account", routeValues: null, htmlAttributes: new { @class = "username", title = "Manage" })!
        @using (Html.BeginForm("LogOff", "Account", FormMethod.Post, new { id = "logoutForm" }))
        {
            @Html.AntiForgeryToken()
            <a href="javascript:document.getElementById('logoutForm').submit()">Log off</a>
        } else {
            <li>@Html.ActionLink("Register", "Register", "Account", routeValues: null, htmlAttributes: new { id = "registerLink" })</li>
            <li>@Html.ActionLink("Login", "Login", "Account", routeValues: null, htmlAttributes: new { id = "loginLink" })</li>
            </ul>
        } else {
    }
</text>

6. Press F5 to run the application. The default ASP.NET MVC application appears in your web browser.
Integrating the Web Application with ACS

1. In Visual Studio 2012, in Solution Explorer, right-click the MvcACS project, and then select Identity and Access.

   If the Identity and Access option does not appear on the menu, install the Identity and Access Tool. For information, see Identity and Access Tool.

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)
2. On the **Providers** tab, select **Use the Windows Azure Access Control Service**.

![Identity and Access](image1.png)

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)

3. Click the **Configure** link.

![Identity and Access](image2.png)

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)
4. Visual Studio requests information about the Access Control namespace. Enter the namespace name you created earlier.

   ![Configure ACS namespace dialog](image)

   *(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)*

   The symmetric key can be found on the Azure portal. Follow these steps:
   
   a. On the Azure Management Portal, in the left pane, click **ACTIVE DIRECTORY**. In the right pane, select an **ACCESS CONTROL NAMESPACE(S)**, and then click **MANAGE**

   ![Azure Management Portal](image)

   *(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)*
b. In the left pane, click **Management service**. Then, in the right pane, click **ManagementClient**.

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)

c. In the **Type** field, select **Symmetric Key**. In the **Key** field, click **Show Key**, and then copy the key value. Then, click **Cancel** to exit the **Edit Management Credential** page without making changes.

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)
5. In Visual Studio, paste the key in the **Enter the management key for the namespace** field, select the **Save management key** option, and then click **OK**.

![Configure ACS namespace](image1)

*(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)*

Visual Studio uses the information about the namespace to connect to the ACS Management Portal and get the settings for your namespace, including the identity providers, realm, and return URL.

6. Select **Windows Live ID** (Microsoft account), and then click **OK**.

![Identity and Access](image2)

*(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)*
Testing the Integration with ACS

Test the integration of your Replying Party Application and ACS.

1. In Visual Studio, press F5 to run the application.

   When your application is integrated with ACS and you have selected Windows Live ID (Microsoft account), instead of opening the default ASP.NET Web Forms application, your browser is redirected to the sign-in page for Microsoft accounts.

2. Sign in with a valid user name and password.

   ![Sign in](image)

   (The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)

You are redirected to the MvcACS application.

![MvcACS](image)

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)
Viewing the Application in the ACS Management Portal

The **Identity and Access** tool in Visual Studio automatically integrates your application with ACS.

When you select the **Use Azure Access Control** option and then run your application, the **Identity and Access** tool adds your application as a relying party, configures it to use the selected identity providers, and generates and selects the default claims transformation rules for the application.

You can review and change these configuration settings in the **ACS Management Portal**. Follow these steps to review the changes in the portal:

2. In the left pane, click **ACTIVE DIRECTORY**. In the right pane, select an **ACCESS CONTROL NAMESPACE**, and then click **MANAGE**.

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)
3. The **ACS Management Portal** is displayed. In the left pane, click **Relying party applications**.

4. The new MvcACS application appears in the list of relying party applications. The realm is automatically set to the application main page. Click **WebACS**.

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)
5. The **Edit Relying Party Application** window contains configuration settings for the MvcACS web application. Change the settings as needed and save them. The changes are immediately applied to the application.

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)

Scroll down the page to see the remaining configuration settings for the MvcACS application, including the identity providers and claims transformation rules.

---

**Token format**
Select a token format for ACS to use when it issues security tokens for this relying party application. Learn more

- **SAML 2.0**

**Token encryption policy**
Select an encryption policy for tokens that ACS issues for this relying party application. Note: Encryption must be selected if the application is a web service that uses proof-of-possession tokens over the WS-Trust protocol because this scenario does not work without encryption. Learn more

- **None**

**Token lifetime (sec)**
Specify the amount of time for a security token that ACS issues to remain valid. Learn more

- **3600**

---

**Authentication Settings**

**Identity providers**
Select the identity providers to use with this relying party application. Learn more

- **Windows Live ID**

**Rule groups**
Select the rule groups to use for this relying party application when processing claims. Learn more

- **W2OACS_RedoGroup**

---

**Token Signing Certificates**

<table>
<thead>
<tr>
<th>Used For</th>
<th>Type</th>
<th>Effective Dates</th>
<th>Status</th>
</tr>
</thead>
</table>

**Token Encryption Certificates**

No token encryption certificates added. To add a token encryption certificate, click here.

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)
Viewing Claims Sent by ACS

It is important to control the claims sent to each application by ACS. To view the claims sent by ACS, modify the application created in “Creating a Web Application to be Secured by ACS” on page 8. The Identity and Access tool has created a rule group that passes through all claims from identity provider to your application. Note that different identity providers send different claims.

1. Open the Controllers\HomeController.cs file. Add a using statement for System.Threading:

   using System.Threading;

2. In the HomeController class, add the Claims method:

   ```csharp
   public ActionResult Claims() { ViewBag.Message = "Your claims page."

   return View();
   }
   ```

3. Right-click on the Claims method and select Add View.

4. Click Add.

   (The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)
5. Replace the content of the **Views\Home\Claims.cshtml** file with the following code:

```html
@{
    ViewBag.Title = “Claims”;
}

<tr>
    @{
        @ViewBag.Title
    }
    <td>
        <h2>@ViewBag.Message</h2>
    </td>
</tr>

<h3>Values from Identity</h3>
<table>
    <tr>
        <td>IsAuthenticated</td>
        <td>@ViewBag.ClaimsIdentity.IsAuthenticated</td>
    </tr>
    <tr>
        <td>Name</td>
        <td>@ViewBag.ClaimsIdentity.Name</td>
    </tr>
</table>

<h3>Claims from ClaimsIdentity</h3>
<table>
    <tr>
        <th>Claim Type</th>
        <th>Claim Value</th>
    </tr>
        <tr>
            <td>@claim.Type</td>
            <td>@claim.Value</td>
        </tr>
    }
</table>
```
6. Press the F5 key to run the application, and then navigate to the **Claims** method.

   ![Image of Claims page]
   
   *(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)*

   For more information on using claims in your application, see the [Windows Identity Foundation library](#).

   **Adding AD FS as an Identity Provider of ACS**

   1. On the ACS Management Portal, click **Identity providers** in the left pane, and then click **Add**.

      ![Image of ACS Management Portal]

      *(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)*
2. Select the **WS-Federation identity provider** option, and then click **Next**.

![Screen Image](image.png)

*(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)*

3. Complete the following fields, and then click **Save**:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Display name</strong></td>
<td>Enter a name (for example, <strong>AD FS 2012</strong>).</td>
</tr>
<tr>
<td><strong>WS-Federation metadata</strong></td>
<td>Select the AD FS metadata (can be download from [http://&lt;AD FS Server&gt;/FederationMetadata/2007-06/FederationMetadata.xml](http://&lt;AD FS Server&gt;/FederationMetadata/2007-06/FederationMetadata.xml)).</td>
</tr>
<tr>
<td><strong>Login Page Settings</strong></td>
<td>Change according to your preferences.</td>
</tr>
<tr>
<td><strong>Used By</strong></td>
<td>Select the relying party applications that will be associated with this identity provider.</td>
</tr>
</tbody>
</table>
4. Ensure that the new identity provider is now added in the list.

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)

5. In Visual Studio, open the project for the MvcACS application, and then click **Identity and Access**. The tool lists both the Windows Live ID and the new **AD FS 2012** identity providers.

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)
Configuring a Trust Relationship Between AD FS and ACS

1. Open the AD FS Management snap-in.
2. In the left pane, click **AD FS > Trust Relationships**, right-click **Relying Party Trusts**, and then click **Add Relying Party Trust**.

   ![AD FS Management snap-in](image1)

   *(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)*

3. On the **Welcome** window in **Add Relying Party Trust Wizard**, click **Start**.

   ![Add Relying Party Trust Wizard](image2)

   *(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)*
4. Now, from the ACS Management portal, copy the ACS WS-Federation Metadata URL.

5. On the Select Data Source window of Add Relying Party Trust Wizard, in the Federation metadata address field, paste the ws-Federation Metadata URL you copied in the previous step. Then, click Next.
6. Keep clicking **Next**, and then finally click **Close** to complete the wizard.
7. If needed, add claim rules as required by your web application.

**Configuring the ACS Relying Party Application’s Rules**

1. On the ACS Management Portal, click **Rule groups** in the left pane.
2. In the right pane, click the web application’s default rule (created automatically when you used the **Identity and Access** tool to secure the web application by ACS).

   ![Rule Groups](image)

   *(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)*

3. On the **Edit Rule Group** window, you can add or edit rules according to the web application’s requirements.

   In our application, User Principal Name (for example, user@domainname.com) received from AD FS needs to be changed to Role, and WindowsAccountName received from AD FS needs to be changed to Name. Click on the rule that has to be changed.

   ![Edit Rule Group](image)

   *(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)*
4. On the **Edit Claim Rule** window, make the changes as required, and then click **Save**.

![](image)

*The screen image above is from Microsoft®. Trademarks are the property of their respective owners.*

5. On the **Edit Rule Group** window, click **Save** to return to home page, and then save the changes.

### Configuring the AD FS Agent

1. Launch the MFA agent.
2. On the **Policy** tab, select **Enable agent** and **Pre-generate challenge**.

![](image)

*The screen image above is from Microsoft®. Trademarks are the property of their respective owners.*
3. On the **Communications** tab, perform the following steps:
   a. In the **Primary Server** field, enter the IP address or name (and the port if non-causal is used) of the SAS server.
   b. If the SAS server is not installed on the same machine as the AD and AD FS, the key encryption file must be loaded. In the **Agent Encryption Key File** field, click **Browse**, and then specify the path of the key encryption file.
      
      The key encryption file was downloaded in step 3 in “A user attempts Sign in to Microsoft Azure Access Control Service. The user is redirected AD FS proxy server (WAP), then after successful authentication, is forwarded to SafeNet Authentication Service (SAS) for a secondary authentication (AD FS multi-factor authentication).

5. The user uses his SAS token for authenticating. SAS collects and evaluates the user’s credentials.

6. The SAS authentication reply is sent back to AD FS which returns a response to Microsoft Azure Access Control Service, accepting or rejecting the user’s authentication request.

7. The user is granted or denied access to Microsoft Azure Access Control Service.

   Configuring a SAS Auth Node and Encryption Key” on page 6.

c. Click **Apply**. Enabling the agent registers the SafeNet MFA adapter with AD FS, and enables it at a global policy level.

d. Verify your settings by testing authentication from the agent to the authentication server. To do so, under **Authentication Test**, enter your **User Name** and **Passcode**, and then click **Test**. The result of the test will be displayed in the **Authentication Test Result** window.

4. **Click OK.**
Configuring the AD FS Authentication Policy

1. On the AD FS Management Console window, in the left pane, click **AD FS > Authentication Policies**.
2. In the far-right pane, click **Edit Global Primary Authentication**.

3. On the **Primary** tab, ensure that **Forms Authentication** is selected for both **Extranet** and **Intranet**.

(The screen image above is from Microsoft® software. Trademarks are the property of their respective owners.)
4. On the Multi-factor tab, perform the following steps:
   a. Under Users/Groups, add the users and/or groups for which MFA will be required.
   b. Under Locations, select Extranet and/or Intranet, according to your preferred configuration.
   c. Ensure that SafeNet Multi Factor Authentication (SMFA) is selected as an additional authentication method.
   d. Click OK.

   ![Edit Global Authentication Policy](image)

   *(The screen image above is from Microsoft® software. Trademarks are the property of their respective owners.)*

### Configuring SafeNet Authentication Service

The deployment of multi-factor authentication using SAS with Microsoft Azure Access Control Service using SAML authentication requires:

- Synchronizing Users Stores to SAS, page 29
- Assigning an Authenticator in SAS, page 30

### Synchronizing Users Stores to SAS

Before SAS can authenticate any user in your organization, you need to create a user store in SAS that reflects the users that would need to use multi-factor authentication. User records are created in the SAS user store using one of the following methods:

- Manually, one user at a time, using the **Create User** shortcut
- Manually, by importing one or more user records via a flat file
- Automatically, by synchronizing with your Active Directory/LDAP server using the SAS Synchronization Agent
For further details on importing users to SafeNet Authentication Service, refer to “Creating Users” in the SafeNet Authentication Service Subscriber Account Operator Guide:


All SafeNet Authentication Service documentation can be found on the SafeNet Knowledge Base site.

Assigning an Authenticator in SAS

SAS supports a number of authentication methods that can be used as a second authentication factor for users authenticating through Microsoft Azure Access Control Service.

The following authenticators are supported:

- eToken PASS
- RB-1 keypad token
- KT-4 token
- SafeNet GOLD
- SMS token
- MP-1 software token
- GrIDsure
- MobilePASS

Authenticators can be assigned to users in two ways:

- **Manual provisioning**—Assign an authenticator to users one at a time.
- **Provisioning rules**—The administrator can set provisioning rules in SAS so that the rules will be triggered when group memberships and other user attributes change. An authenticator will be assigned automatically to the user.

Refer to “provisioning” in the SafeNet Authentication Service - Subscriber Account Operator Guide to learn how to provision the different authentication methods to the users in the SAS user store.

Running the Solution

For this integration, the SafeNet GrIDsure token is configured for authentication with the SAS solution.

1. Browse to your web application.
2. You are redirected to the ACS login page. Click the new **AD FS 2012** relying party.

3. You are redirected to the SAS login page. Enter your organizational credentials, and then click **Sign in**.

(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)
4. The challenge grid is displayed. In the **OTP** field, enter the characters from the grid that correspond to your PIP (personal identification pattern), and then click **Submit**.

![SafeNet Authentication Service](image)

After successful authentication, you are logged in to your target web application.

![Screen Image](image)

*(The screen image above is from Microsoft®. Trademarks are the property of their respective owners.)*
Support Contacts

If you encounter a problem while installing, registering, or operating this product, please make sure that you have read the documentation. If you cannot resolve the issue, contact your supplier or Gemalto Customer Support. Gemalto Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Gemalto and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

<table>
<thead>
<tr>
<th>Contact Method</th>
<th>Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Address</strong></td>
<td>Gemalto, Inc.</td>
</tr>
<tr>
<td></td>
<td>4690 Millennium Drive</td>
</tr>
<tr>
<td></td>
<td>Belcamp, Maryland  21017 USA</td>
</tr>
<tr>
<td><strong>Phone</strong></td>
<td>United States</td>
</tr>
<tr>
<td></td>
<td>1-800-545-6608</td>
</tr>
<tr>
<td></td>
<td>International</td>
</tr>
<tr>
<td></td>
<td>1-410-931-7520</td>
</tr>
<tr>
<td><strong>Technical Support</strong></td>
<td><a href="https://serviceportal.safenet-inc.com">https://serviceportal.safenet-inc.com</a></td>
</tr>
<tr>
<td><strong>Customer Portal</strong></td>
<td>Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the Gemalto Knowledge Base.</td>
</tr>
</tbody>
</table>