Key rollover on ADFS

This page describes the process of certificate rollover for ADFS Identity Providers. The procedure described below allows replacing certificates without any service disruptions.

In SWAMID default installation we have both an Encryption and a Signing certificate.

Info : Certificate automatic rollover

ADFS default setting is to use Certificate automatic rollover.
This means that ADFS will create new certificates and roll them at according to its own schedule.

When does this happen? In the *Get-AdfsProperties* command, you can check the value for *CertificateCriticalThreshold*.

The default setting is 2 and it means that ADFS will switch the certificates two days before their expiration date weather you want it to or not.

The next parameter of interest is *CertificatePromotionThreshold*, the default value of 5 means the old certificate will be present as a secondary certificate for five days after rollover.

We recommend to turn off automatic certificate rollover to ensure that you have control of the process.

Check if automatic certificate rollover is turned on with the following command:

```
Get-AdfsProperties | Select AutoCertificateRollover
```

To turn it off, use the following command:

```
Set-AdfsProperties -AutoCertificateRollover $false
```

Step 1 : Create new certificate

- Use the following commands to create the encryption and signing certificates. Change the values needed.

```
$ADFSDnsName = "[ADFS DNS Name]"

# Encryption Certificate
$encryptionCertificate = New-SelfSignedCertificate -CertStoreLocation "cert:\LocalMachine\My" -DnsName
"encryption.$ADFSDnsName" -KeyExportPolicy Exportable -KeyLength 4096 -NotAfter (Get-Date).AddYears(10) -KeySpec
KeyExchange

# Signing Certificate
$signingCertificate = New-SelfSignedCertificate -CertStoreLocation "cert:\LocalMachine\My" -DnsName
"signing.$ADFSDnsName" -KeyExportPolicy Exportable -KeyLength 4096 -NotAfter (Get-Date).AddYears(10) -KeySpec
KeyExchange
```

Step 2 : Export certificates and import on all ADFS servers in the farm

- Export the certificates on the server where you created them. Change FilePath if needed

```
# Encryption Certificate
$encryptionPassword = Read-Host "Type a secure password for the encryption PFX file" -AsSecureString
Export-PfxCertificate -Cert $encryptionCertificate -FilePath ("C:\{0}.pfx" -f $ADFSDnsName -Password $password

# Signing Certificate
$signingPassword = Read-Host "Type a secure password for the signing PFX file" -AsSecureString
Export-PfxCertificate -Cert $signingCertificate -FilePath ("C:\{0}.pfx" -f $ADFSDnsName -Password $password
```

- Copy the certificates to all ADFS servers in the farm and import them.

```
$ADFSDnsName = "[ADFS DNS Name]"

# Encryption Certificate
$encryptionPassword = Read-Host "Type the previous password for the encryption PFX file" -AsSecureString
Import-PfxCertificate -FilePath 'C:\encryption.ADFS DNS Name.pfx' -Exportable:$true -Password $encryptionPassword
-CertStoreLocation "cert:\LocalMachine\My"

# Signing Certificate
$signingPassword = Read-Host "Type the password for the signing PFX file" -AsSecureString
Import-PfxCertificate -FilePath 'C:\signing.ADFS DNS Name.pfx' -Exportable:$true -Password $signingPassword -
CertStoreLocation "cert:\LocalMachine\My"
```

Step 3 : Give ADFS service account access to the private key

- Start the Certificate MMC for the local machine

```
certlm.msc
```

- Navigate to Personal Certificates
- For each new certificate, right click and select All tasks Manage Private Keys...
- Add the service account for the ADFS service and click OK
- Do this on all ADFS servers in the farm

Step 4 : Add Certificates into the ADFS database

- Go back to the server where you created the certificate(s)
- Run the following command to import the certificates into the ADFS database
  The certificates will be imported as secondary certificates, no services will be affected.

```
# Encryption Certificate
Add-AdfsCertificate -CertificateType "Token-Decrypting" -Thumbprint $encryptionCertificate.Thumbprint

# Signing Certificate
Add-AdfsCertificate -CertificateType "Token-Signing" -Thumbprint $signingCertificate.Thumbprint
```

Step 5 : Upload new Metadata

- Upload the XML from https://adfs.example.com/federationmetadata/2007-06/federationmetadata.xml to metadata.swamid.se/admin
- Remove the SP / IdP part unless ADFS will be used as both roles
- Use "Merge missing from published" to copy over all EntityCategory's and MDUI information from the old Entity
- Request publication
- Wait until you get confirmation of publication and then wait for at least 8h (recommended 24h if the IdP is only in SWAMID and 48h if the IdP is in eduGAIN) to make sure all entities have picked up the new certificate(s)

Step 6 : Certificate rollover

This can be done in the console by right clicking the certificate and select "Set as primary" or by PowerShell:

```
# Encryption Certificate
Set-AdfsCertificate -CertificateType "Token-Decrypting" -Thumbprint "[thumbprint of cert]" -IsPrimary

# Signing Certificate
Set-AdfsCertificate -CertificateType "Token-Signing" -Thumbprint "[thumbprint of cert]" -IsPrimary
```

Step 7 : Upload new Metadata

We need to publish the new metadata where the new certificates are primary.

- Follow step 5 to upload the new metadata to metadata.swamid.se/admin
- Remove the secondary certificate from the web tool

Step 8 : Remove the old certificates

This can be done in the console by right clicking the certificate and select "Delete" or by PowerShell:

```
# Encryption Certificate
Remove-AdfsCertificate -CertificateType "Token-Decrypting" -Thumbprint "[thumbprint of cert]"

# Signing Certificate
Remove-AdfsCertificate -CertificateType "Token-Signing" -Thumbprint "[thumbprint of cert]"
```

The certificates are still in the Local machine certificate store. Delete them from the Certificate MMC.

```
certlm.msc
```