Fortanix®

Integration Guide

USING FORTANIX DATA SECURITY MANAGER WITH SECTIGO FOR CODESIGNING

VERSION 1.0

Fortanix Inc. | 800 West El Camino Real | Suite 180 | Mountain View, CA 94040 | United States of America | +1 (650) 943-2484 |
info@fortanix.com | www.fortanix.com

1.0 INTRODUCTION

This article describes how to integrate **Fortanix Data Security Manager (DSM) with Sectigo for code signing**. It also contains the information that a user requires to:

- Prepare the Build Server / Code-Signing Workstation
- Configure Fortanix Data Security Manager for Sectigo CA Code-Signing integration
- Generate/Import Private Key and Certificate

1.1 OVERVIEW

Fortanix DSM has state-of-the-art code signing solution that offers the following capabilities:

1. FIPS 140-2 level 3 assurance for private key protection.
2. Supports all types of asymmetric keys, signing, and hashing algorithms used for code signing. It also supports signing just the hash.
3. Code signing in large enterprises often requires verification of metadata associated with the data being signed as well as access control around the use of keys. These checks can easily be performed in a secure environment using plugins in Fortanix DSM.
4. Code signing keys are very sensitive, and their use should be tightly controlled. Fortanix DSM provides elaborate quorum-based policies to be configured for these keys which require approval from M of N administrators before the signing operation is performed. These approvals can be obtained in an asynchronous and distributed fashion.
5. Strict role-based-access-control, quorum-based approval workflows, automation, and audit logs for all code signing operations.
6. Support of 100% for REST APIs, KMIP, PKCS11, JCE, Microsoft CAPI, and CNG for easy integration with your existing DevOps tooling.
7. Code signing is future proof in Fortanix DSM. Post-quantum algorithms, such as LMS, are already supported and can be used for code signing.

2.0 PREPARING THE BUILD SERVER / CODE-SIGNING WORKSTATION

The Server/Workstation that will be running the SignTool must have the following installed:

- Fortanix DSM CNG Provider:
    - Link: https://support.fortanix.com/hc/en-us/articles/360018084132-CNG-EKM
    - Once installed, validate that the provider has been correctly registered.

```
C:\> certutil -csplist
```

FIGURE 1: VALIDATION

- SignTool:
  - SignTool is now part of Windows SDK and is required.
  - Link: https://developer.microsoft.com/en-us/windows/downloads/windows-10-sdk/

3.0 FORTANIX DATA SECURITY MANAGER CONFIGURATION

Fortanix DSM will require appropriate groups and apps to be pre-created before the integration begins.

1. Create an appropriate group that will be managing the security objects within the account.

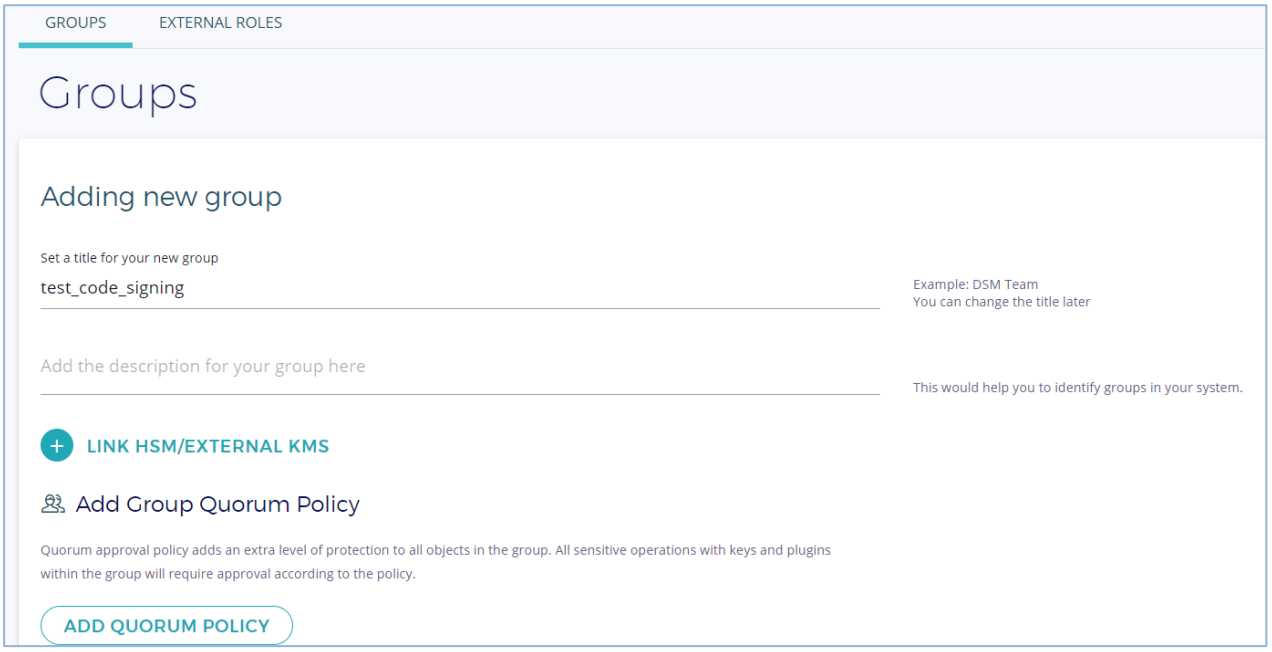

FIGURE 2: CREATE GROUP

2. Create a new app in Fortanix DSM that will provide an API Key that will be used to authenticate when communicating using CNG provider (take note of the API Key).

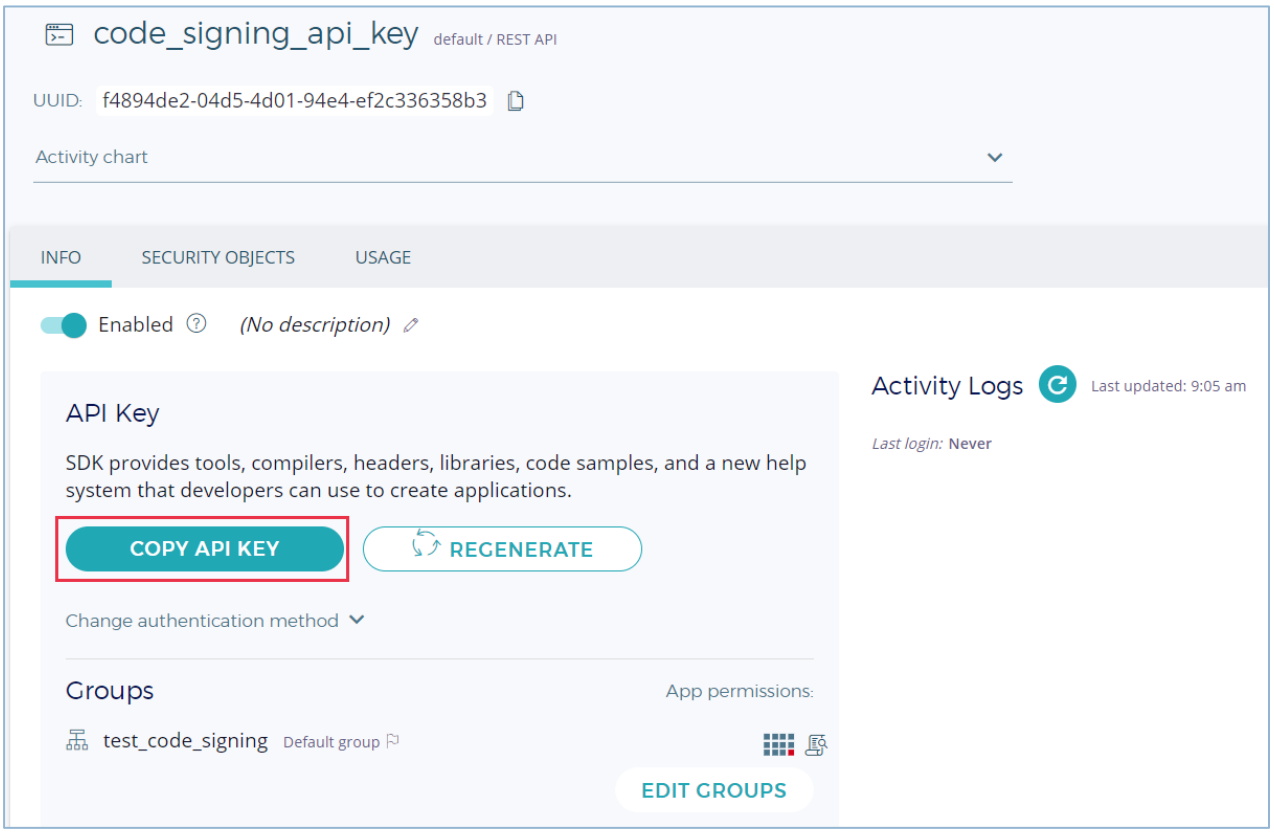

FIGURE 3: CREATE APP AND COPY API KEY

3. On the Build Server/Code-Signing Workstation, Fortanix DSM CNG Provider requires couple of configuration variables, which will be stored in the registry.
   - Fortanix DSM Endpoint
   - Fortanix DSM API Key

   The following two commands will store the correct values in the registry (you may also choose a user to store the entries in your user registry instead of HKLM (HKEY_LOCAL_MACHINE).

```
C:\Program Files\Fortanix\KmsClient>FortanixKMSClientConfig.exe machi
ne --api-endpoint https://amer.smartkey.io
```

```
C:\Program Files\Fortanix\KmsClient>FortanixKMSClientConfig.exe machi
ne --api-key ZGZiNzc0OGMtYmM0Mi00NGYzLTgxNTEtNTYyMzMxOTAxMmVjOkZDSjAx
VS1nRHJHc0lYd1FaanZ4dktid0U2ei16M0VneTBGRWtzQnJfYUNwY3RRcUhXalhQcHZqe
DZzRzB4ZzNkRmkzb0x2ZVMtcm9uSlJRVFlpRXFB
```

4. Confirm Fortanix KMS CNG Provider can communicate properly with Fortanix DSM.

```
C:\Program Files\Fortanix\KmsClient>certutil -csp "Fortanix KMS CNG Provider" -key
Fortanix KMS CNG Provider:
  sectigo_private_key
  9a7f801b-37b5-4341-a8b4-4f3e244f30a4
  RSA
```

FIGURE 4: CONFIRM THE COMMUNICATION

4.0 GENERATE OR IMPORT THE PRIVATE KEY AND CERTIFICATE

Securing the Private Keys and Certificates are the most critical tasks to ensure codes cannot be maliciously signed by offending parties. Fortanix supports generating/importing and securing the appropriate security objects:

- Generate the Private Key using Fortanix DSM UI, create a Certificate Sign Request from SignTool, and then import the Certificate into Fortanix DSM once signed by Sectigo (trusted Certificate Authority).

4.1 GENERATE PRIVATE KEY ON FORTANIX DATA SECURITY MANAGER / GENERATE CSR THROUGH `certreq.exe`

This method will generate the Private Key and Certificate sign request from Fortanix DSM and `certreq.exe`. Upon receiving a signed certificate from the trusted Certificate Authority, the certificate can then be imported into Fortanix DSM.

1. Create a new security object that will be the Private Key and assign to the appropriate group (in this example, we will call the security object – **`sectigo_private_key`**):

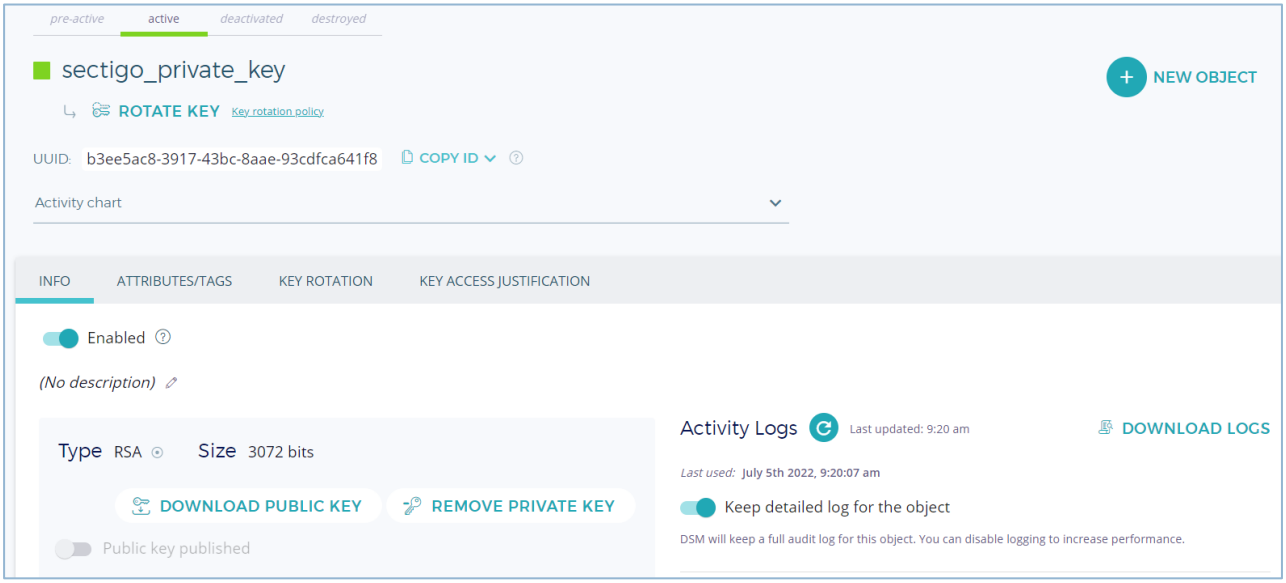

FIGURE 5: CREATE NEW SECURITY OBJECT

OR you can also generate the key using PowerShell:

```
$cngProviderName    = "Fortanix KMS CNG Provider"
$cngAlgorithmName   = "RSA"
$cngKeySize         = <size-of-RSA-Key>              # Recommended key size for column
master keys
$cngKeyName         = "<name-of-security-object>"             # Name identifying your key
in the KSP
$cngProvider        = New-Object System.Security.Cryptography.CngProvider($cngProviderName)

$cngKeyParameter    = [System.Security.Cryptography.CngKeyCreationParameters]::new()
$cngKeyParameter.Provider = $cngProvider
$cngKeyParameter.KeyCreationOptions =
[System.Security.Cryptography.CngKeyCreationOptions]::MachineKey


$keySizeProperty = New-Object System.Security.Cryptography.CngProperty("Length",
[System.BitConverter]::GetBytes($cngKeySize),
[System.Security.Cryptography.CngPropertyOptions]::None)
$cngKeyParameter.Parameters.Add($keySizeProperty)

$cngAlgorithm = New-Object System.Security.Cryptography.CngAlgorithm($cngAlgorithmName)
$cngKey = [System.Security.Cryptography.CngKey]::Create($cngAlgorithm, $cngKeyName,
$cngKeyParameter)
```

After the PowerShell command runs, you will notice that the key has been created in the Fortanix DSM UI.

2. Generate the Certificate Sign Request using the private key using the `CETREQ.EXE`:
    a. Create a new file called `request.inf` in a temporary directory.
    b. Replace the following content into the file:
        i. **`KeyContainer`**: Name of the security object created previously/Private Key.
        ii. **`ProviderName`**: Based on the provider name when installing the Fortanix CNG Provider.

```
[NewRequest]

Subject = "CN=sectigo_private_key, OU=nishank, O=Fortanix, C=US"
KeyContainer = "sectigo_private_key"
ProviderName = "Fortanix KMS CNG Provider"
UseExistingKeySet = true

[EnhancedKeyUsageExtension]
OID=1.3.6.1.5.5.7.3.3
```

    c. Type the following command to generate the Certificate Sign Request.

```
C:\> certreq.exe -new request.inf request.csr
```

d. This command will now generate a `request.csr` Certificate Sign Request file and should be sent to the trusted Certificate Authority to receive a signed Certificate. Sample file:


```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIEiTCCAvECAQAwUDELMAkGA1UEBhMCVVMxETAPBgNVBAoMCEZvcnRhbml4MRAw
DgYDVQQLDAduaXNoYW5rMRwwGgYDVQQDDBNzZWN0aWdvX3ByaXZhdGVfa2V5MIIB
ojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEAq5lA0CUztBzLWV0kkgH/qk94
CcOKZODm1LC8b3NF/pZuEWtUd1sryQVuOVbK3upiuthyMnsUNrADM+YwUf0iuxUb
e4EKJM4at5rUjE2nq7hzuWmR7LVDzCFniOwtxTOjAL7kViMKPlayzMkzJ/Lswx4k
ei4zGnxjpoUR0wBIMEIFL+FyHyDLlBWXfQ4/9H+BDOvC2KY+FUg8Co+cwUkPUkqV
VzT7n9IChA/tMgzwaRjcERjt6lcrtaWWnSdWreWCoH0iWfHheej//+wU6gNJMoNh
Hsn3yKAiIGmcmCqYeU/o92uzLQfLuNqCrXlMk2zdiyOL66wonoyANyhU2z/3HzHG
kaA0ETR01Kl3K+D4y9ovJCbagWbVX56SqQkPM2i2tT9MLRcMa24ao8MZJQ+GOIVh
or/4gvvVt+Dll/mCKWhxq3SgG9Xoaog2M3yP8A5UhzPlzmYWmq/XueZtSykRF0do
WoDI6RnSTCrT9ETDH3zhNjMIgRU1R0RS7ZlSWMzVAgMBAAGggfMwHAYKKwYBBAGCNw0CAzEOFgwxMC4wLjE5MDQ0LjIwQgYJKwYBBAGCNxUUMTUwMwIBCQwPTEFQVE9QLVJCSlFJMzgwDBRBenVyZUFEXE5pc2hhbmtWYWlzaAwHY2VydHJlcTBDBgkqhkiG9w0BCQ4xNjA0MBMGA1UdJQQMMAoGCCsGAQUFBwMDMB0GA1UdDgQWBBQJhe3z15cbLap0W1VSJSPoeUq46DBKBgorBgEEAYI3DQICMTwwOgIBAB4yAEYAbwByAHQAYQBuAGkAeAAgAEsATQBTACAAQwBOAEcAIABQAHIAbwB2AGkAZABlAHIDAQAwDQYJKoZIhvcNAQELBQADggGBAAMOVhOHvhpeLOblRtVhra0apYb/ACt4w1598RdzJn34yuajRt+F7eSweQ+msylnUZkzNeDYAtD4tXerLnM0A11Xv9L1oQjghpng0ZDzSCvk94DRunzVfjFxDSv6pgJ9VzxljAm0rkCB/x6vF6t9wd5PaEm0odAnxCZYEvCEYQfloyft
HxLk6gqN5k5fIPMtkt2kjOy55DKS83FbIe1LeCapI+0SADQLV6jNHjd5PEJfiBPb
Nq9bMM88DwOfWino/s+5RZKmDdcETOf78VrkszKvckMYp/CMnHuVrM3GpOpuBQQMl4FcqToddNPlNnecRh0oA7Ecxaqa+xBj3DGGG4jGonMsW5qMYjMk3aI6SwWgiFuHfXk5EmN+RyK+lW/b5Ogy1VKW4ERLdhjWtMqCVqsv3s9inUtx7Jc9Wb1ynR2j4Qzz
21AOVGgJgjp3R1gbuudl8M6CGE+ekGO+BZwpcDQAsV4Tu5j+53yPgnFunGgTDYfS
zoakzNCyRjpbBn3BAw==
-----END NEW CERTIFICATE REQUEST-----
```


3. Purchase a Code Signing Certificate from Sectigo and log in to your Sectigo Client Dashboard.

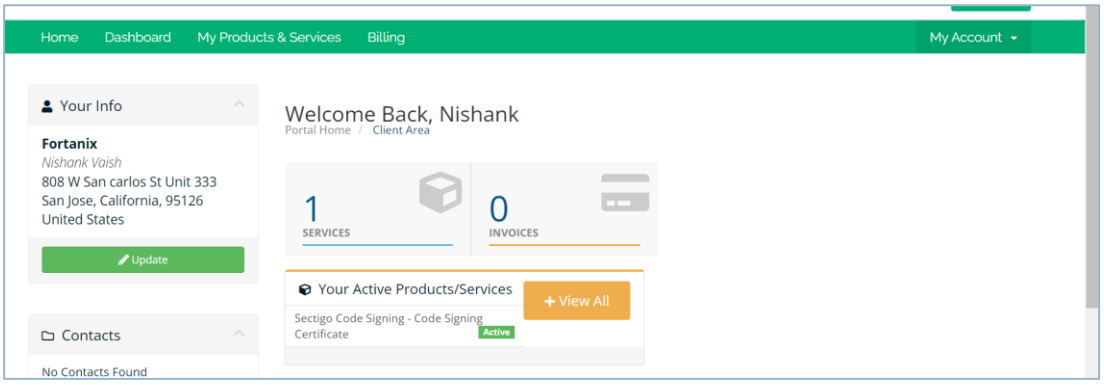

FIGURE 6: SECTIGO CLIENT DASHBOARD

4. Click your "Active" product and request your certificate.

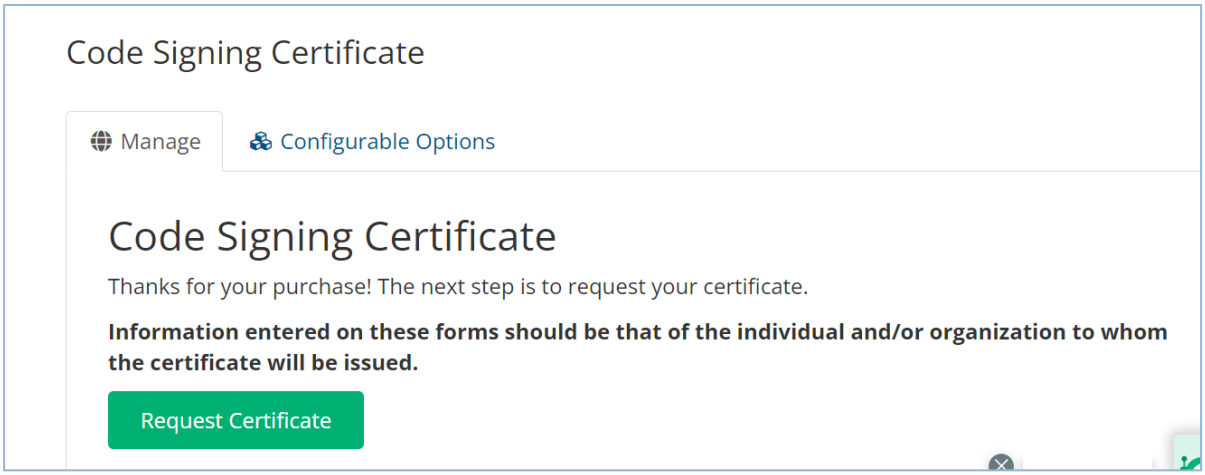

FIGURE 7: REQUEST CERTIFICATE

5. Copy-paste the `request.CSR` file which was created in *Step 2(d)* above and submit the certificate request.

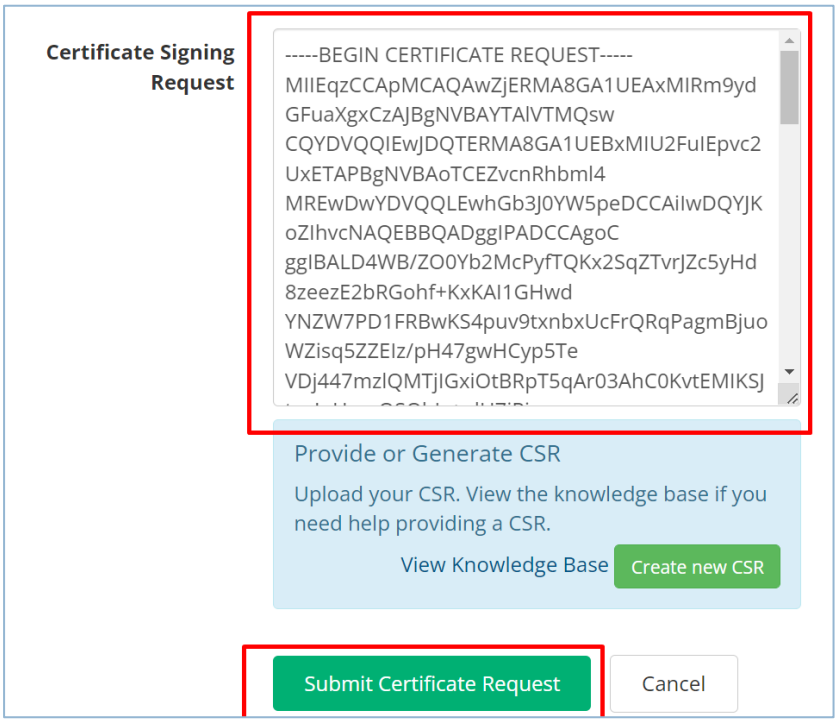

FIGURE 8: SUBMIT CERTIFICATE REQUEST

6. Once the signed Certificate is received, you can import the certificate into Fortanix DSM.

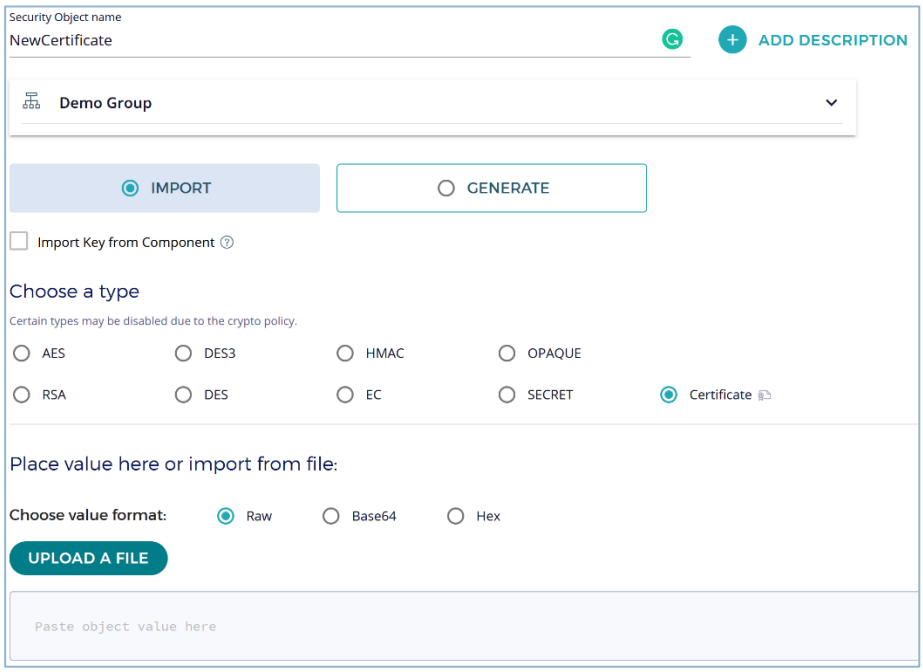

FIGURE 9: IMPORT SIGNED CERTIFICATE

7. Keep a copy of the certificate on the server where the SignTool will be run from (the certificate can be exported from Fortanix DSM at any time).

5.0 CODE-SIGNING INTEGRATION (DIRECTLY FROM WORKSTATION)

1. Verify no other signatures are present on the file that will be signed.

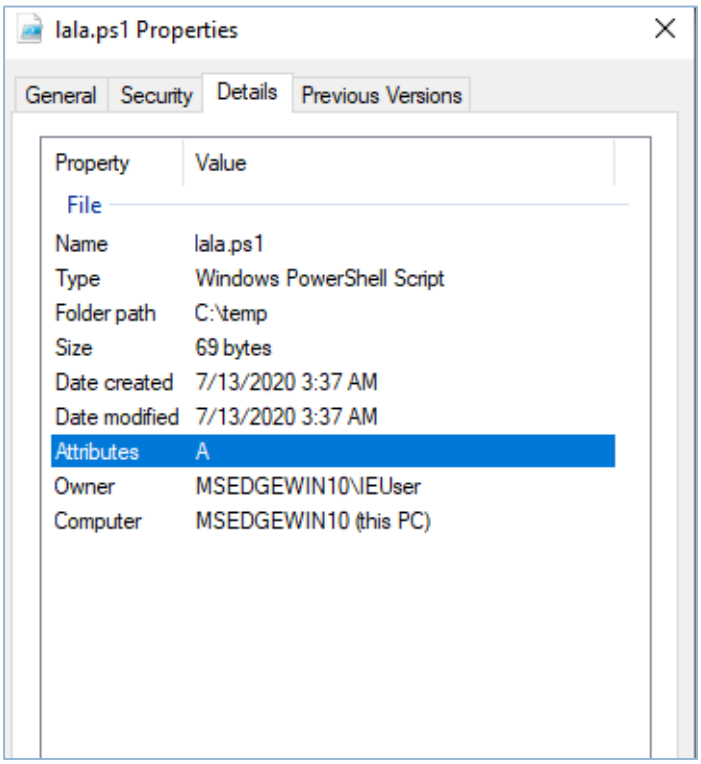

FIGURE 10: VERIFY SIGNATURE

2. Open a command prompt. Locate the file SignTool that is appropriate for your code (for example: x64, x86, and so on).

```
C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0>dir
 Volume in drive C is Windows 10
 Volume Serial Number is B4A6-FEC6

 Directory of C:\Program Files (x86)\Windows Kits\10\bin\10.0.19041.0

07/13/2020  01:24 AM    <DIR>          .
07/13/2020  01:24 AM    <DIR>          ..
07/13/2020  01:24 AM    <DIR>          arm
07/13/2020  01:24 AM    <DIR>          arm64
07/13/2020  04:18 AM    <DIR>          x64
07/13/2020  01:24 AM    <DIR>          x86
               0 File(s)              0 bytes
               6 Dir(s)  22,404,153,344 bytes free
```

FIGURE 11: LOCATE SIGNTOOL

3. Verify that the key you wish to use to sign the code is available in the remote CNG provider.

```
C:\Program Files\Fortanix\KmsClient>certutil -csp "Fortanix KMS CNG Provider" -key
Fortanix KMS CNG Provider:
  sectigo_private_key
  9a7f801b-37b5-4341-a8b4-4f3e244f30a4
  RSA
```

FIGURE 12: VERIFY THE KEY

4. The following command will sign the code specified in the SignTool and require the following parameters at a minimum to successfully run the SignTool:
   - **CSP**: The CNG provider you wish to use for the sign operation.
   - **KC**: Key Container (also known as an alias) that will be used for the sign operation.
   - **File**: Certificate generated from the Private Key stored in Fortanix DSM.
   - **Code to sign**.

```
C:\> signtool.exe sign /csp "Fortanix KMS CNG Provider" ^
     /kc "Sectigo_private_key"/f c:\temp\request.cer c:\temp\lala.ps1
Done Adding Additional Store
Successfully signed: c:\temp\lala.ps1
```

FIGURE 13: SIGN THE CODE

If using a certificate already stored in the certstore, you may also omit the CSP and KC parameters:

```
C:\> signtool.exe sign /sha1 <thumbprint-of-cert> C:\temp\lala.ps1
Done Adding Additional Store
Successfully signed: c:\temp\lala.ps1
```

FIGURE 14: OMIT THE CSP AND KC

5. Once the file has been signed, Fortanix DSM will log an event within the audit log to signify the private key was used to sign the code.

6.0 FREQUENTLY ASKED QUESTIONS

1. How do I validate the supported algorithms and modes using Fortanix KMS CNG Provider?

- You can view all of the supported methods, algorithms, and modes with Fortanix DSM using the CNG provider by running a `csptest`:

```
certutil -csp "Fortanix KMS CNG Provider" -csptest

Provider Name: Fortanix KMS CNG Provider
  Impl Type: 1 (0x1)
    NCRYPT_IMPL_HARDWARE_FLAG -- 1

  Version: 196610 (0x30002)
  Pass

  Provider Aliases:
    Fortanix KMS CSP

  Provider Module:
      UM(1): FortanixKmsCngProvider.dll
      0(1): 10001, 1
        0: KEY_STORAGE

  Asymmetric Encryption Algorithms:
   RSA
    BCRYPT_ASYMMETRIC_ENCRYPTION_INTERFACE -- 3
    NCRYPT_ASYMMETRIC_ENCRYPTION_OPERATION -- 4
    NCRYPT_SIGNATURE_OPERATION -- 10 (16)

  Signature Algorithms:
   RSA
    BCRYPT_ASYMMETRIC_ENCRYPTION_INTERFACE -- 3
    NCRYPT_ASYMMETRIC_ENCRYPTION_OPERATION – 4
...
    CAPI_KDF
    TLS1_1_KDF
    TLS1_2_KDF
    HKDF

CertUtil: -csptest command completed successfully.
```

FIGURE 15: VALIDATE SUPPORTED ALGORITHMS

7.0 DOCUMENT INFORMATION

7.1 DOCUMENT LOCATION

The latest published version of this document is located at the URL :

https://support.fortanix.com/hc/en-us/articles/7535961661076-Using-Fortanix-Data-Security-Manager-with-Sectigo-CA-for-Code-Signing

7.2 DOCUMENT UPDATES

This document will typically be updated on a periodic review and update cycle.

For any urgent document updates, please send an email to: support@fortanix.com