Cybersource for Salesforce B2C Commerce REST-Message-Level Encryption Guide

Contents

Introduction

This document provides a step-by-step guide for managing P12 certificates and code changes for MLE implementation without upgrading the cartridge. It covers the process of generating a P12 file, extracting the necessary certificates, and importing them into Business Manager. Additionally, it outlines the essential code modifications required to ensure seamless integration and functionality.

Steps for managing P12 certificate

Follow the below steps to create, extract and import the P12 certificate.

Step 1: Create P12 file

1. Follow steps mentioned in the link to generate a P12 certificate in Business Center.
2. Make a note of password set to the P12 key.
3. Download the generated P12 file.

Step 2: Extract Certificate from the P12 file

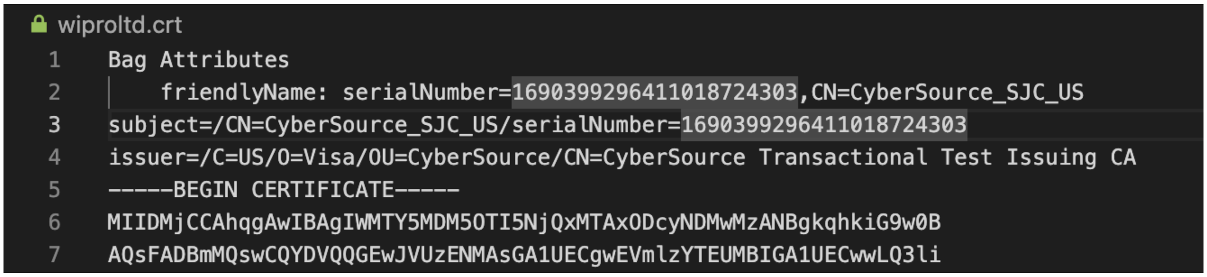
Extract the Certificate Authority (CA) Certificate from the downloaded P12 file using OpenSSL tool.

1. Open the terminal in the folder where the P12 file is stored.
2. Run the below OpenSSL command to extract the CA certificates from the downloaded P12 file.

```
openssl pkcs12 -in <Merchant_ID>.p12 -cacerts -nokeys -out <Merchant_ID>.crt
```

Note: Replace the **<Merchant_ID>** with the merchant ID for which you wish to configure.

3. A new file named **<Merchant_ID>.crt** will be generated in the same directory as the P12 file.
4. Note the Serial Number of the **"Cybersource_SJC_US"** certificate.
   Refer the below example:



wiproltd.crt

```
Bag Attributes
    friendlyName: serialNumber=1690399296411018724303,CN=CyberSource_SJC_US
subject=/CN=CyberSource_SJC_US/serialNumber=1690399296411018724303
issuer=/C=US/O=Visa/OU=CyberSource/CN=CyberSource Transactional Test Issuing CA
-----BEGIN CERTIFICATE-----
MIIDMjCCAhqgAwIBAgIWMTY5MDM5OTI5NjQxMTAxODcyNDMwMzANBgkqhkiG9w0B
AQsFADBmMQswCQYDVQQGEwJVUzENMAsGA1UECgwEVmlzYTEUMBIGA1UECwwLQ3li
```

5. Update the Serial Number to the custom preferences at the following path:
   Merchant Tools > Site Preferences > Custom Site Preference Groups > Message-Level Encryption Configuration

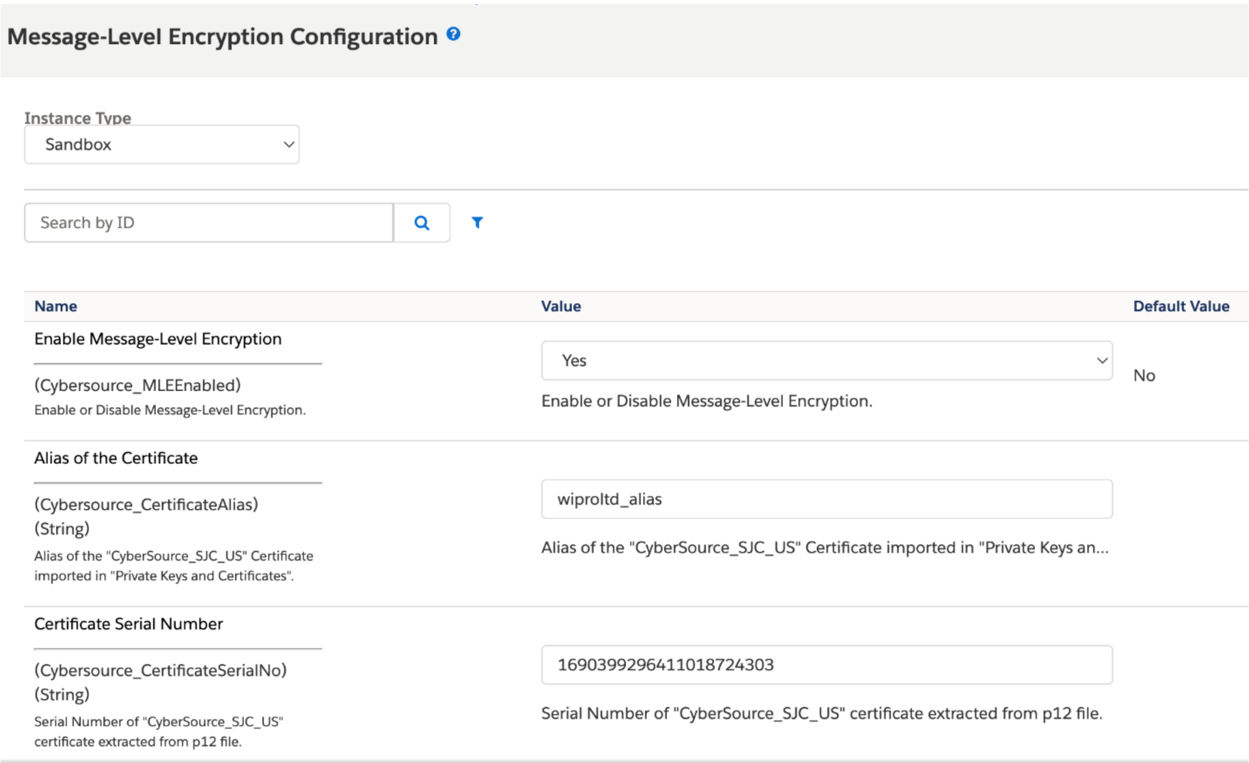
Step 3: Import the extracted certificate file in Business Manager

1. Import the extracted **<Merchant_ID>.crt** certificate file in Business Manager at the following path:
   Administration > Operations > Private Keys and Certificates

2. Use any alias for the certificate and update the same alias in the custom preferences at the following path:

Merchant Tools > Site Preferences > Custom Site Preference Groups > Message-Level Encryption Configuration

Refer to the below example for MLE configuration in custom preferences:



Message-Level Encryption Configuration

Instance Type: Sandbox

Name	Value	Default Value
Enable Message-Level Encryption (Cybersource_MLEEnabled) Enable or Disable Message-Level Encryption.	Yes Enable or Disable Message-Level Encryption.	No
Alias of the Certificate (Cybersource_CertificateAlias) (String) Alias of the "CyberSource_SJC_US" Certificate imported in "Private Keys and Certificates".	wiproltd_alias Alias of the "CyberSource_SJC_US" Certificate imported in "Private Keys an...	
Certificate Serial Number (Cybersource_CertificateSerialNo) (String) Serial Number of "CyberSource_SJC_US" certificate extracted from p12 file.	1690399296411018724303 Serial Number of "CyberSource_SJC_US" certificate extracted from p12 file.	

Code changes for MLE

Step 1: Add following files in the cartridge

Navigate to the path **“cartridges/int_cybs_sfra_base/cartridge/scripts/mleEncrypt”** in our cartridge version 25.2.0 available on GitHub. Here, there are two files named **aesgcmCustom.js** and **jweEncrypt.js**. Add these files to your custom cartridge. These two files are used to encrypt the request payload.

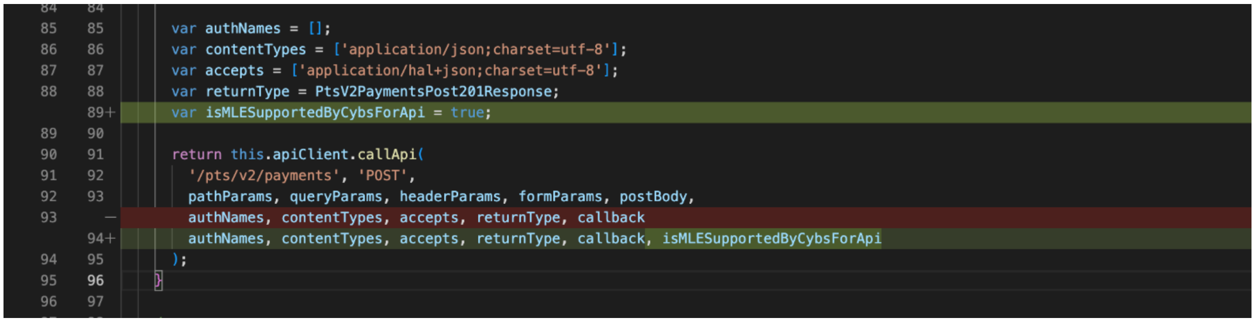
Step 2: Changes in API files

Update the following code changes to **all the functions** in **“int_cybs_sfra_base/cartridge/apiClient/api/PaymentsApi.js”** and **“int_cybs_sfra_base/cartridge/apiClient/api/PayerAuthenticationApi.js”** file.

The changes will be same for all the functions in these two files.

Declare the following variable and add the same variable to the **callApi()** function call as argument, similar to the screenshot below.

```
var isMLESupportedByCybsForApi = true;
```



```diff
     var authNames = [];
     var contentTypes = ['application/json;charset=utf-8'];
     var accepts = ['application/hal+json;charset=utf-8'];
     var returnType = PtsV2PaymentsPost201Response;
+    var isMLESupportedByCybsForApi = true;

     return this.apiClient.callApi(
       '/pts/v2/payments', 'POST',
       pathParams, queryParams, headerParams, formParams, postBody,
-      authNames, contentTypes, accepts, returnType, callback
+      authNames, contentTypes, accepts, returnType, callback, isMLESupportedByCybsForApi
     );
   }
```

Step 3: Changes in ApiClient.js file

Add following changes in **"int_cybs_sfra_base/cartridge/apiClient/ApiClient.js"** file.

1. Import the index file as below.

```
var configObject = require('*/cartridge/configuration/index');
```

2. Update callApi function by adding **"isMLESupportedByCybsForApi"** parameter.

```
_exports.prototype.callApi = function (path, httpMethod, pathParams, queryParams,
headerParams, formParams, bodyParam, authNames, contentTypes, accepts,
returnType, callback, isMLESupportedByCybsForApi) {
```

3. Add the following changes in **callApi**() function after line 220
   i.e., after line payload = JSON.stringify(bodyParam);

```
var isMLEEnabled = configObject.mleEnabled;

if (isMLEEnabled && isMLESupportedByCybsForApi == true) {
    var encryptPayload = require('*/cartridge/scripts/mleEncrypt/jweEncrypt.js');
    payload = encryptPayload.getJWE(payload);
}
```

Step 4: Make changes to read the MLE configurations from Business Manager

1. Add below lines in the **"int_cybs_sfra_base/cartridge/configuration/preferences/customPreferences.js"** file.

```
/* MLE Custom Preference */
 MLE: {
    id: 'Cybersource_MLE',
    display_name: 'Message-Level Encryption Configration',
    Preferences: {
        /** @type {CustomPreference} */
        EnableMLE: {
            id: 'Cybersource_MLEEnabled',
            display_name: 'Enable Message-Level Encryption',
            description: 'Enable or Disable Message-Level Encryption.',
            type: Types.boolean,
            default: false,
            flags: {
                mandatory: false
            }
        },
        /** @type {CustomPreference} */
        MLECertificateSerialNumber: {
            id: 'Cybersource_CertificateSerialNo',
            display_name: 'Certificate Serial Number',
            description: 'Serial Number of "CyberSource_SJC_US" certificate
extracted from p12 file.',
            type: Types.string,
            default: undefined,
            flags: {
                mandatory: false
            }
        },
         /** @type {CustomPreference} */
         MLECertificateAlias: {
            id: 'Cybersource_CertificateAlias',
            display_name: 'Alias of the Certificate',
            description: '',
            type: Types.string,
            default: undefined,
            flags: {
                mandatory: false
            }
        }
    }
},
```

2. Add the following lines in **"int_cybs_sfra_base/cartridge/configuration/index.js"** file in **getConfig()** function.

```
//MLE
mleEnabled: config.mleEnabled ||
customPreferences.MLE.Preferences.EnableMLE.getValue(),
mleCertificateSerialNumber: config.mleCertificateSerialNumber ||
customPreferences.MLE.Preferences.MLECertificateSerialNumber.getValue(),
mleCertificateAlias: config.mleCertificateAlias ||
customPreferences.MLE.Preferences.MLECertificateAlias.getValue()
```

Step 5: Metadata changes to create configurations

1. Create a new file named **MLE.xml** at path **"metadata/payments_metadata/meta"** and add the following lines of code to the file.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<metadata xmlns="http://www.demandware.com/xml/impex/metadata/2006-10-31">
    <type-extension type-id="SitePreferences">
        <custom-attribute-definitions>
            <attribute-definition attribute-id="Cybersource_MLEEnabled">
                <display-name xml:lang="x-default">Enable Message-Level
Encryption</display-name>
                <description xml:lang="x-default">Enable or Disable Message-Level
Encryption.</description>
                <type>boolean</type>
                <externally-managed-flag>false</externally-managed-flag>
                <default-value>false</default-value>
            </attribute-definition>
            <attribute-definition attribute-id="Cybersource_CertificateSerialNo">
                <display-name xml:lang="x-default">Certificate Serial
Number</display-name>
                <description xml:lang="x-default">Serial Number of
"CyberSource_SJC_US" certificate
                    extracted from p12 file.</description>
                <type>string</type>
                <externally-managed-flag>false</externally-managed-flag>
            </attribute-definition>
            <attribute-definition attribute-id="Cybersource_CertificateAlias">
                <display-name xml:lang="x-default">Alias of the
Certificate</display-name>
                <description xml:lang="x-default">Alias of the
"CyberSource_SJC_US" Certificate
                    imported in "Private Keys and Certificates".</description>
                <type>string</type>
                <externally-managed-flag>false</externally-managed-flag>
            </attribute-definition>
        </custom-attribute-definitions>
        <group-definitions>
            <attribute-group group-id="Cybersource_MLE">
                <display-name xml:lang="x-default">Message-Level Encryption
Configuration</display-name>
                <attribute attribute-id="Cybersource_MLEEnabled" />
                <attribute attribute-id="Cybersource_CertificateSerialNo" />
                <attribute attribute-id="Cybersource_CertificateAlias" />
            </attribute-group>
        </group-definitions>
    </type-extension>
</metadata>
```

2. Make changes to **“metadata/payments_metadata/meta/merged.xml”** file:

Add following lines of code in <custom-attribute-definitions> element of the xml.

```xml
<attribute-definition attribute-id="Cybersource_MLEEnabled">
    <display-name xml:lang="x-default">Enable Message-Level Encryption</display-
name>
    <description xml:lang="x-default">Enable or Disable Message-Level
Encryption.</description>
    <type>boolean</type>
    <externally-managed-flag>false</externally-managed-flag>
    <default-value>false</default-value>
</attribute-definition>
<attribute-definition attribute-id="Cybersource_CertificateSerialNo">
    <display-name xml:lang="x-default">Certificate Serial Number</display-name>
    <description xml:lang="x-default">Serial Number of "CyberSource_SJC_US"
certificate extracted
        from p12 file.</description>
    <type>string</type>
    <externally-managed-flag>false</externally-managed-flag>
</attribute-definition>
<attribute-definition attribute-id="Cybersource_CertificateAlias">
    <display-name xml:lang="x-default">Alias of the Certificate</display-name>
    <description xml:lang="x-default">Alias of the "CyberSource_SJC_US"
Certificate imported in
        "Private Keys and Certificates".</description>
    <type>string</type>
    <externally-managed-flag>false</externally-managed-flag>
</attribute-definition>
```

Add following lines of code in <group-definitions> element of the xml.

```xml
<attribute-group group-id="Cybersource_MLE">
  <display-name xml:lang="x-default">Message-Level Encryption
Configuration</display-name>
  <attribute attribute-id="Cybersource_MLEEnabled" />
  <attribute attribute-id="Cybersource_CertificateSerialNo" />
  <attribute attribute-id="Cybersource_CertificateAlias" />
</attribute-group>
```