OPEN BANKING

OpenSSL

eIDAS PSD2 Certificate Signing Request Profiles

Date: 26th July 2019

Issue: 2.1

Classification: Public

References			
Number	**Title**	**Link**	**Notes**
1	Electronic Signatures and Infrastructures (ESI); Sector Specific Requirements; Qualified Certificate Profiles and TSP Policy Requirements under the payment services Directive (EU) 2015/2366”	https://www.etsi.org/deliver/etsi_ts/119400_119499/119495/01.02.01_60/ts_119495v010201p.pdf	ETSI specification for QWAC and QSEAL certificates.

TABLE OF CONTENTS

1 PREFACE....4

1.1 Purpose of this document....4

2 X509 CERTIFICATE SIGNING REQUEST PROFILES....5

2.1 OBWAC and OBSEAL....5

2.2 PSD2 Statement....5

2.3 Organisational Identifier statement....5

2.4 CSR generation - Openssl....6

2.5 ASN.1 Editor....7

2.6 Validation of Qcstatement der encoding....8

3 OBWAC X509 CERTIFICATE SIGNING REQUEST PROFILE....9

3.1 Introduction....9

3.2 OBWAC OpenSSL CNF file....9

3.3 PSP_PI Role....13

3.4 PSP_PI and PSP_AI Roles....13

3.5 Combinations of all PSP_XX roles: PSP_AS + PSP_PI + PSP_AI + PSP_IC 14

3.6 Example OBWAC CSR creation for roles PSP_PI, PSP_AI and PSP_IC....15

4 OBSEAL X509 CERTIFICATE SIGNING REQUEST PROFILE....18

4.1 Introduction....18

4.2 OBSEAL OpenSSL CNF file....18

4.3 PSP_PI Role....22

4.4 PSP_PI and PSP_AI Roles....22

4.5 Combinations of all PSP_XX roles: PSP_AS + PSP_PI + PSP_AI + PSP_IC 23

4.6 Example OBSEAL CSR creation for roles PSP_AS and PSP_IC....24

1 PREFACE

1.1 PURPOSE OF THIS DOCUMENT

This document outlines the general procedure to generate Certificate Signing Requests (CSR) for Open Banking Public Key Infrastructure (PKI).

In particular, it concentrates on the method required to generate a CSR compliant with ETSI specification “Electronic Signatures and Infrastructures (ESI); Sector Specific Requirements; Qualified Certificate Profiles and TSP Policy Requirements under the payment services Directive (EU) 2015/2366” **[REF 1]**

2 X509 CERTIFICATE SIGNING REQUEST PROFILES

2.1 OBWAC AND OBSEAL

The OBWAC and/or OBSEAL certificates are signed by the Open Banking PKI once a client has generated a valid CSR. The ETSI specification **[REF 1]** requires specific, formatted entries for certain x509 certificate fields. In particular:

- QcStatement of type PSD2. One or more PSD2 statements MUST be in the certificate.
- QcStatement of type QCType. MUST be either QcType Web or QcType Seal.
- Organisational Identifier MUST be in subject name.

Both OBWAC and OBSEAL must contain these fields. In addition, OBWAC certificates are issued with the extended key usage field of id-kp-serverAuth and id-kp-clientAuth [RFC5280].

The QcStatements must be encoded correctly in the CSR submitted by the client. This following sections details an approach to do this.

Generally, QCStatement may optionally contain other QCStatement types defined by ETSI
such as QcCompliance, QcLimitValue etc. However, these QcStatement types are not mandatory for OBWAC or OBSEAL type certificates and are not discussed further.

2.2 PSD2 STATEMENT

As specified in [**REF 1]**

The PSD2 specific attributes shall be included in a QCStatement within the qcStatements extension as specified in clause 3.2.6 of IETF RFC 3739 [7].

The QCStatement shall contain the following PSD2 specific certificate attributes as required by RTS [i.3] article 34:

- *The role of the payment service provider, which maybe one or more of the following:*
  - *account servicing (PSP_AS);*
  - *payment initiation (PSP_PI);*
  - *account information (PSP_AI);*
  - *issuing of card-based payment instruments (PSP_IC).*
- *The name of the competent authority where the payment service provider is registered. This is provided in two forms:*
  - *the full name string (NCAName) in English and*
  - *an abbreviated unique identifier (NCAId).*

Generally, to encode the above information into a CSR requires generating a valid ASN.1 representation. This is discussed later in the document.

2.3 ORGANISATIONAL IDENTIFIER STATEMENT

As specified in [**REF 1]**

The PSD2 Authorization Number, or other identifier recognized by the NCA, shall be placed in organizationIdentifier attribute of the Subject Distinguished Name field in the certificate.

The organizationIdentifier "PSDGB-OB-Unknown0015800001041ReAAI" means a certificate issued to a TPP where the authorization number is Unknown0015800001041ReAAI, authorization was granted by UK Open Banking (identifier PSDGB-OB).

Other examples can include use of non-alphanumeric characters such as "PSDBE-NBB-1234.567.890", "PSDFI-FINFSA-1234567-8" and "PSDMT-MFSA-A 12345" (note space character after "A").

After consultations between European Banking Authority (EBA) and ETSI it was found that there are institutions which can request PSD2 certificates but have no authorization number. If the authorization number was not issued by the NCA, then another registration identifier recognized by the NCA is used from those defined in the standard ETSI EN/TS 319 412-1

2.4 CSR GENERATION - OPENSSL

There are many utilities to generate CSR's to submit to a certificate authority. This document uses OpenSSL to demonstrate how a valid CSR can is generated to submit to Open Banking to issue an OBWAC or OBSEAL x509 certificate that complies with ETSI certificate template.

OpenSSL does not offer an ASN.1 editor to natively generate PSD2 type statements out of the box although it does offer an ASN.1 parser. OpenSSL is supported on Windows and Unix platforms and the client is free to generate a CSR on the platform of their choice. **The client is reminded to protect the private key according to their security policies**.

OpenSSL does support embedding a valid DER encoded hexadecimal representation of a PSD2 statement in a CSR. Therefore, another tool is required to actually generate a valid PSD2 statement, which is DER, encoded.

This document is not intended to be a tutorial on OpenSSL. However, general usage is based on installing OpenSSL on your target operating system and generating a CSR as follows:

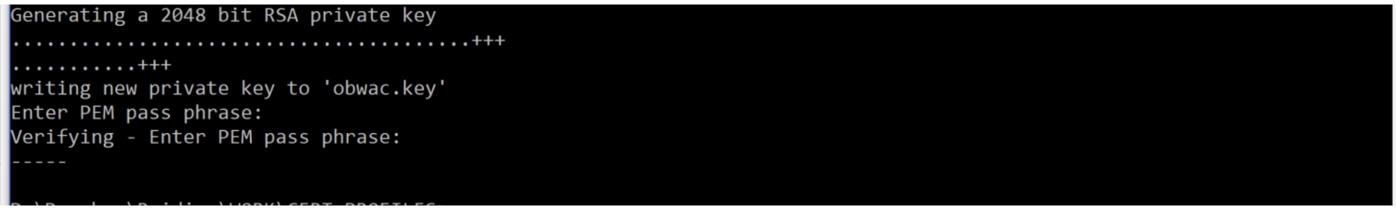
OBWAC CSR Generation

# openssl req -new -config obwac.cnf -out obwac.csr -keyout obwac.key

OBSEAL CSR Generation

# openssl req -new -config obseal.cnf -out obsealc.sr -keyout obseal.key

Example:



```
Generating a 2048 bit RSA private key
.....................................+++
...........+++
writing new private key to 'obwac.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
```

The obwac.key and/or obseal.key files must be protected by your security policies.

The obwac.csr and/or obseal.csr are submitted to Open Banking PKI for processing.

If successful, a signed certificate is returned associated with the appropriate key file generated above.

The contents of the obwac.cnf and obseal.cnf are dependent on what type of PSD2 service provider the client generating the CSR is, such as a PSP_AS .

Example OpenSSL cnf files are supplied for clients in the next chapter.

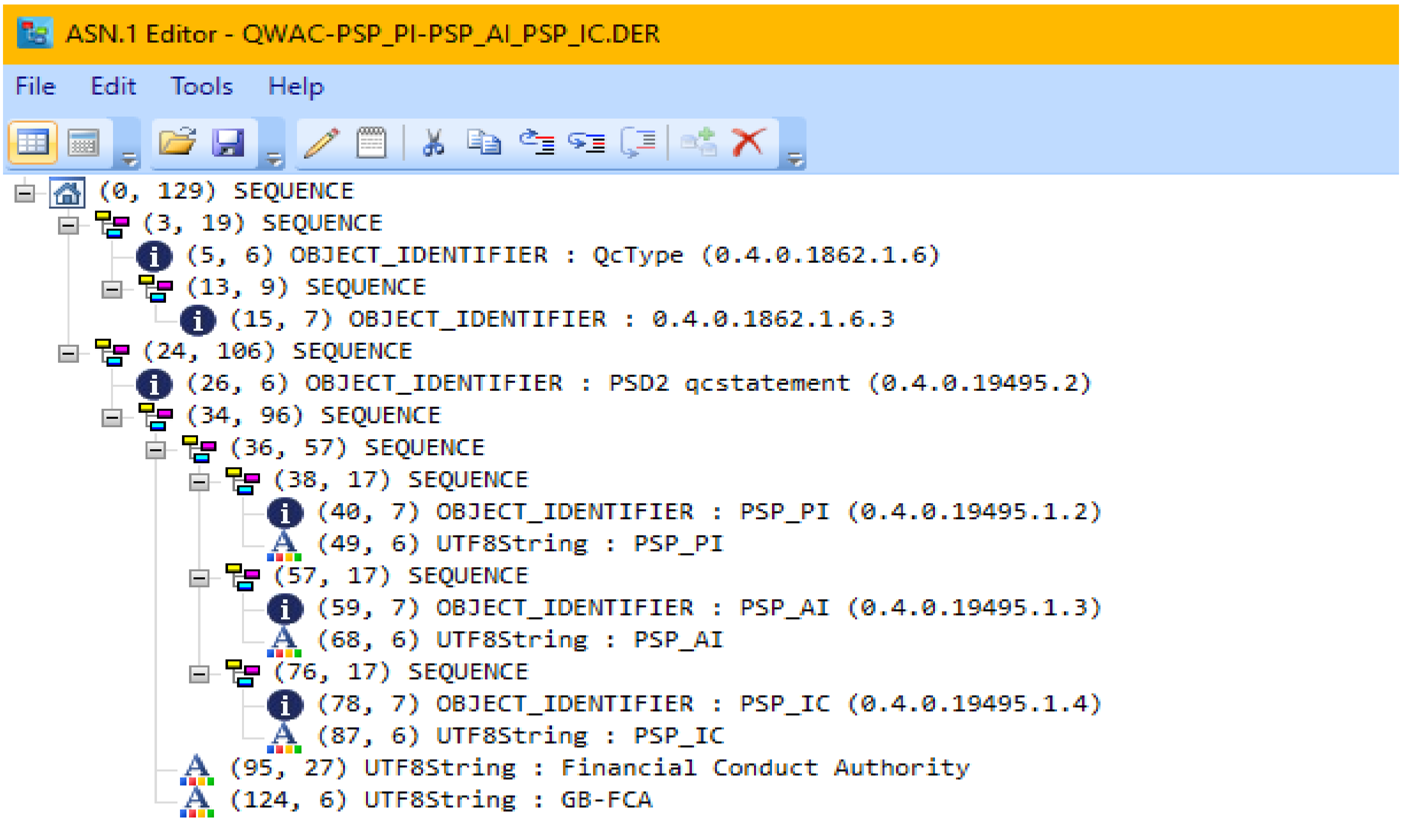
2.5 ASN.1 EDITOR

Vadims Podāns has created an excellent ASN.1 editor and parser. It is ONLY available on Microsoft Windows. However, the output can be used in OpenSSL cnf files on Windows or Unix

The following URL contains link to download the tool:

https://www.sysadmins.lv/projects/asn1editor/default.aspx

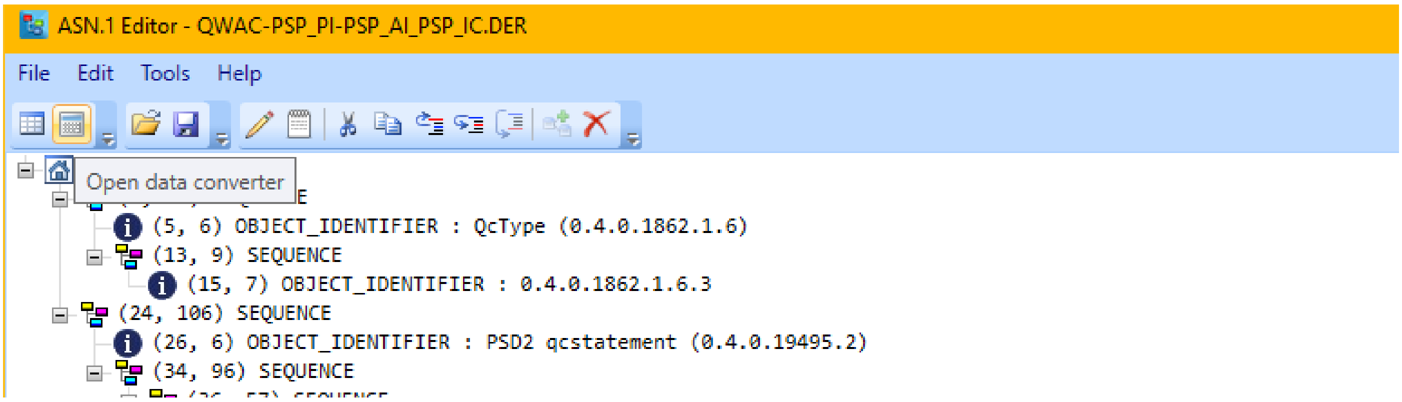
ASN.1 Editor is a tool that allows you to display, edit, format and convert ASN.1-encoded data.



```
(0, 129) SEQUENCE
  (3, 19) SEQUENCE
    (5, 6) OBJECT_IDENTIFIER : QcType (0.4.0.1862.1.6)
    (13, 9) SEQUENCE
      (15, 7) OBJECT_IDENTIFIER : 0.4.0.1862.1.6.3
  (24, 106) SEQUENCE
    (26, 6) OBJECT_IDENTIFIER : PSD2 qcstatement (0.4.0.19495.2)
    (34, 96) SEQUENCE
      (36, 57) SEQUENCE
        (38, 17) SEQUENCE
          (40, 7) OBJECT_IDENTIFIER : PSP_PI (0.4.0.19495.1.2)
          (49, 6) UTF8String : PSP_PI
        (57, 17) SEQUENCE
          (59, 7) OBJECT_IDENTIFIER : PSP_AI (0.4.0.19495.1.3)
          (68, 6) UTF8String : PSP_AI
        (76, 17) SEQUENCE
          (78, 7) OBJECT_IDENTIFIER : PSP_IC (0.4.0.19495.1.4)
          (87, 6) UTF8String : PSP_IC
      (95, 27) UTF8String : Financial Conduct Authority
      (124, 6) UTF8String : GB-FCA
```

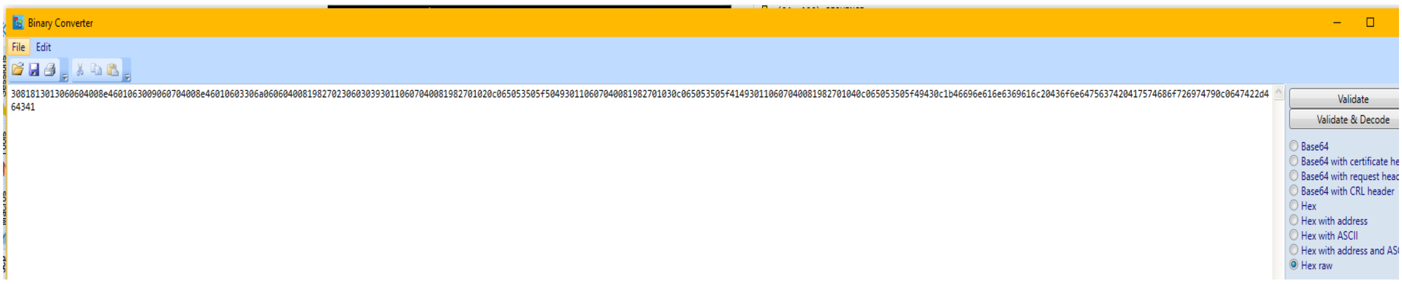
The tools allows a Qualified Certificate Statement with a PSD2 statement [OID= 0.4.0.19495.2] to be constructed as shown above. In the example, the organisation has a PSD2 roles of PSP_PI, PSP_AI, PSP_IC, a nCAName of Financial Conduct Authority and a nCAId of GB-FCA.

The above example also demonstrates a Qualified Certificate Statement with a Qualified Type of QcWeb (OID= 0.4.0.1862.1.6.3]. This indicates it is an OBWAC certificate.



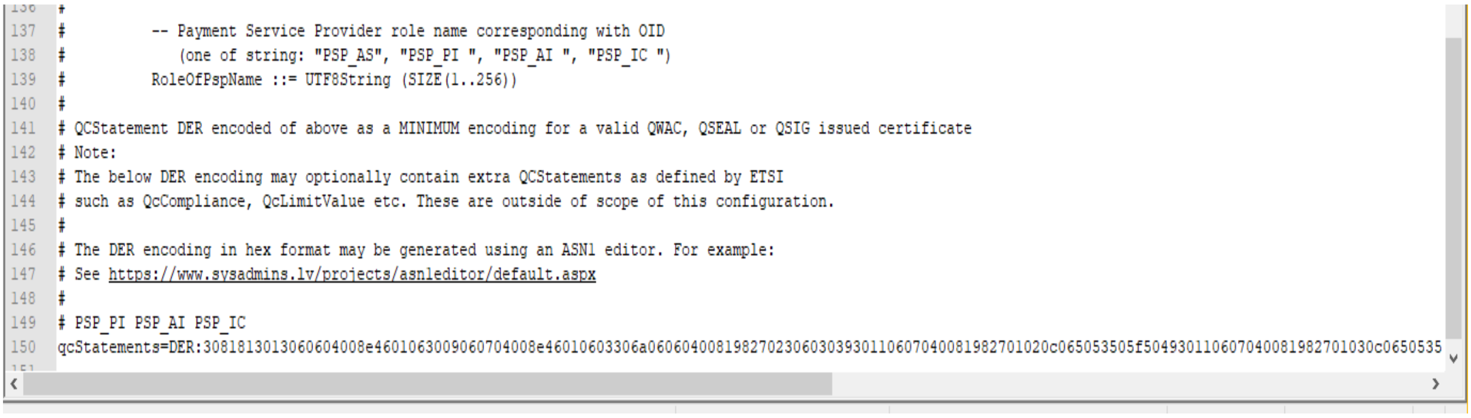
```
    (5, 6) OBJECT_IDENTIFIER : QcType (0.4.0.1862.1.6)
    (13, 9) SEQUENCE
      (15, 7) OBJECT_IDENTIFIER : 0.4.0.1862.1.6.3
  (24, 106) SEQUENCE
    (26, 6) OBJECT_IDENTIFIER : PSD2 qcstatement (0.4.0.19495.2)
    (34, 96) SEQUENCE
```

The raw, hexadecimal encoding of the above settings are generated by pressing the “Open data converter” button as shown above.



```
3081813013060604008e4601063009060704008e46010603306a06060400819827023060303930110607040081982701020c065053505f50493011060704008198270103 0c065053505f414930110607040081982701040c065053505f49430c1b46696e616e6369616c20436f6e647563742041757468f726974790c0647422d4
64341
```

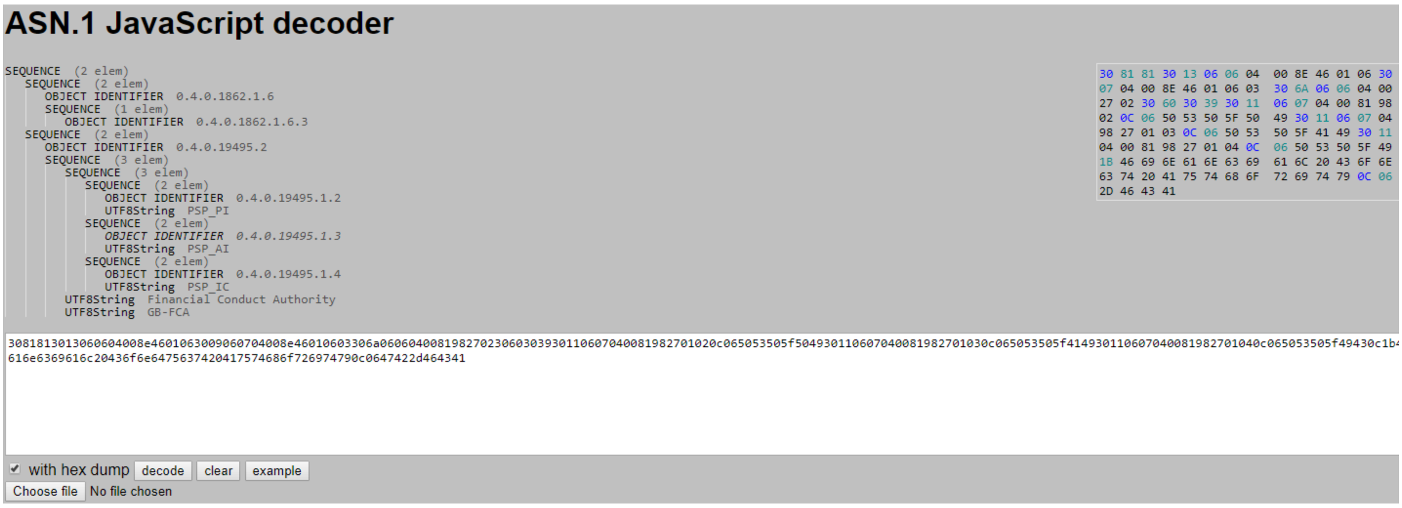
Selecting the “Hex Raw” radio button generates a long line of hexadecimal as shown above.



```
#
#         -- Payment Service Provider role name corresponding with OID
#            (one of string: "PSP_AS", "PSP_PI ", "PSP_AI ", "PSP_IC ")
#         RoleOfPspName ::= UTF8String (SIZE(1..256))
#
# QCStatement DER encoded of above as a MINIMUM encoding for a valid QWAC, QSEAL or QSIG issued certificate
# Note:
# The below DER encoding may optionally contain extra QCStatements as defined by ETSI
# such as QcCompliance, QcLimitValue etc. These are outside of scope of this configuration.
#
# The DER encoding in hex format may be generated using an ASN1 editor. For example:
# See https://www.sysadmins.lv/projects/asnleditor/default.aspx
#
# PSP_PI PSP_AI PSP_IC
qcStatements=DER:3081813013060604008e4601063009060704008e46010603306a06060400819827023060303930110607040081982701020c065053505f50493011060704008198270103 0c0650535
```

The hexadecimal string can be copied into the appropriate OpenSSL obwac.cnf or obseal.cnf file as shown above. This allows generation of a CSR with the appropriate encoding of client details.

2.6 VALIDATION OF QCSTATEMENT DER ENCODING



ASN.1 JavaScript decoder

```
SEQUENCE  (2 elem)
   SEQUENCE  (2 elem)
      OBJECT IDENTIFIER  0.4.0.1862.1.6
      SEQUENCE  (1 elem)
         OBJECT IDENTIFIER  0.4.0.1862.1.6.3
   SEQUENCE  (2 elem)
      OBJECT IDENTIFIER  0.4.0.19495.2
      SEQUENCE  (3 elem)
         SEQUENCE  (3 elem)
            SEQUENCE  (2 elem)
               OBJECT IDENTIFIER  0.4.0.19495.1.2
               UTF8String  PSP_PI
            SEQUENCE  (2 elem)
               OBJECT IDENTIFIER  0.4.0.19495.1.3
               UTF8String  PSP_AI
            SEQUENCE  (2 elem)
               OBJECT IDENTIFIER  0.4.0.19495.1.4
               UTF8String  PSP_IC
         UTF8String  Financial Conduct Authority
         UTF8String  GB-FCA
```

```
30 81 81 30 13 06 06 04  00 8E 46 01 06 30
07 04 00 8E 46 01 06 03  30 6A 06 06 04 00
27 02 30 60 30 39 30 11  06 07 04 00 81 98
02 0C 06 50 53 50 5F 50  49 30 11 06 07 04
98 27 01 03 0C 06 50 53  50 5F 41 49 30 11
04 00 81 98 27 01 04 0C  06 50 53 50 5F 49
1B 46 69 6E 61 6E 63 69  61 6C 20 43 6F 6E
63 74 20 41 75 74 68 6F  72 69 74 79 0C 06
2D 46 43 41
```

```
3081813013060604008e4601063009060704008e46010603306a06060400819827023060303930110607040081982701020c065053505f504930110607040081982701030c065053505f414930110607040081982701040c065053505f49430c1b4
616e6369616c20436f6e647563742041757468f726974790c0647422d464341
```

The following web site contains an ASN.1 DER decoder to validate the DER encoding is as expected. If not using OpenSSL, this is useful to look at the DER encoding before incorporating in the CSR.

https://lapo.it/asn1js

3 OBWAC X509 CERTIFICATE SIGNING REQUEST PROFILE

3.1 INTRODUCTION

This chapter demonstrates how to generate a valid CSR by a client that requires an OBWAC x509 certificate issued to them.

3.2 OBWAC OPENSSL CNF FILE

The following configuration file is used to generate a CSR for an OBWAC certificate using OpenSSL.

All text highlighted in red must be changed by the client to specific data pertinent to them.

NB.

A client can use the OpenSSL cnf file below to generate a CSR which is subsequently signed using their own certificate authority. However, such a certificate will not be trusted by Open Banking.

obwac.cnf file for OpenSSL CSR request

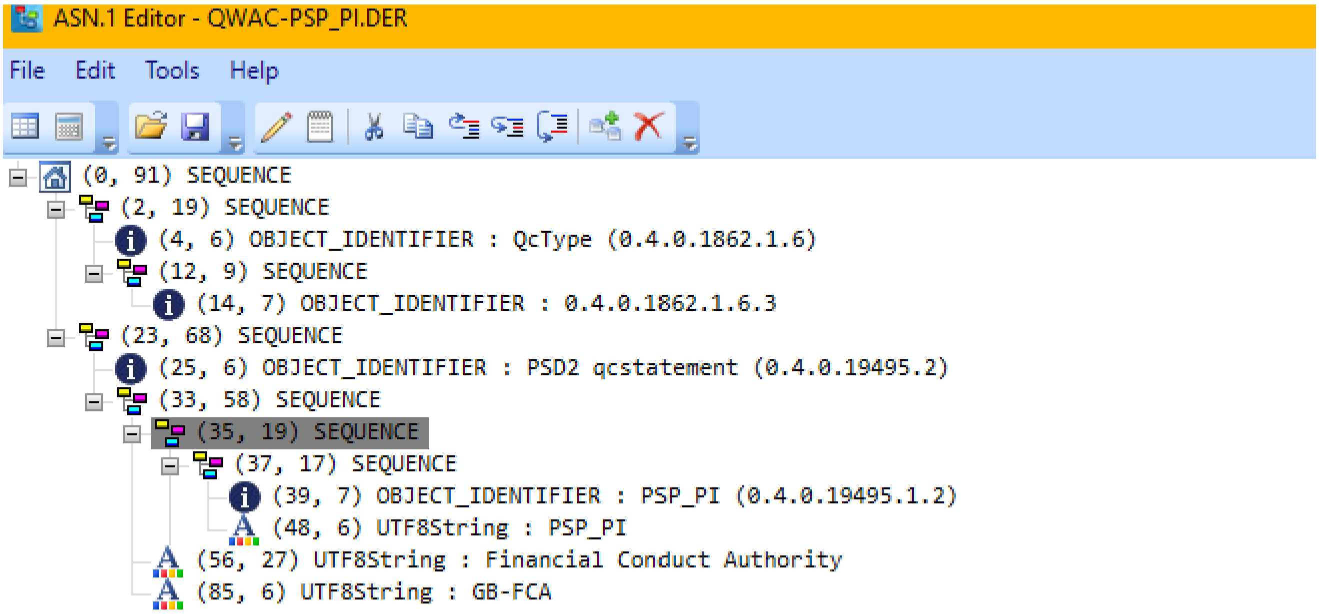
```
#
# OPENSSL CSR REQUEST CONFIGURATION FILE
# ======================================
#
# OBWAC qualified client certificate request with PSD2 role: PSP_PI PSP_AI
# ------------------------------------------------------------------------
# See latest specification: ETSI TS 119 495 V1.2.1 (2018-11)
# https://www.etsi.org/deliver/etsi_ts/119400_119499/119495/01.02.01_60/ts_119495v010201p.pdf
#
oid_section            = new_oids

[ new_oids ]
organizationIdentifier  = 2.5.4.97              # OpenSSL may not recognize this OID so need to add.

[ req ]
default_bits            = 2048                  # RSA key size
encrypt_key             = yes                   # Protect private key: yes or no. yes recommended
default_md              = sha256                # MD to use. sha256 recommended
utf8                    = yes                   # Input is UTF-8.
string_mask             = utf8only              # Emit UTF-8 strings
prompt                  = no                    # Prompt for DN. yes or no.
distinguished_name      = client_dn         # DN template. Mandatory to include organizationIdentifier
req_extensions          = client_reqext         # Desired extensions. Mandatory to include PSD2 QCStatements

#
# Subject Distinguished Name format in certificate
# -----------------------------------------------
# EG: CN = 0015800001041ReAAI, 2.5.4.97 = PSDGB-OB-Unknown0015800001041ReAAI, O = Open Banking Limited (D), C = GB
#
#
[ client_dn ]
countryName             = "GB"                         # Country code - see doc above
organizationName        = "Open Banking Limited (D)"   # Organizational name
#
#  organizationIdentifier
#  ----------------------
#  The organizationIdentifier shall be present in the Subject's Distinguished Name
#  and encoded with legal person syntax
#
#  EXAMPLE: The organizationIdentifier "PSDPL-PFSA-1234567890" means a certificate issued to a PSP where
#  the authorization number is 1234567890, authorization was granted by the Polish Financial
#  Supervision Authority (identifier after second hyphen-minus is decided by Polish numbering
#  system). Other examples can include use of non-alphanumeric characters such as "PSDBE-NBB-
#  1234.567.890" and "PSDFI-FINFSA-1234567-8" and "PSDMT-MFSA-A 12345" (note space character after "A")
#
organizationIdentifier  = "PSDGB-OB-Unknown0015800001041ReAAI"      # Must be in format as shown above
commonName              = "0015800001041ReAAI"                      # Subject common name
#
# Required specific extensions in certificate
#
[ client_reqext ]
keyUsage                = critical,digitalSignature                 # Must be critical
#
# NOTE: As stated in section 7.1.2.3 item f of the CA/Browser Forum Baseline Requirements [i.8]
# (as referenced in ETSI EN 319 412-4 [4]) "id-kp-serverAuth or id-kp-clientAuth [RFC5280] or both values
#  MUST be present". If the certificate is intended to be used as the client certificate in
#  mutual authentication then both values of extKeyUsage certificate extension will need to be present.
#  It is not intended that certificates issued under this profile are used just as client certificates.
#
extendedKeyUsage        = clientAuth, serverAuth                    # Must be defined as shown above
subjectKeyIdentifier    = hash                                      # Hash value to calculate SKI
#
#
#   QC-STATEMENT
#
#      FROM PKIXqualified97 {iso(1) identified-organization(3) dod(6)
#      internet(1) security(5) mechanisms(5) pkix(7) id-mod(0) id-mod-qualified-cert-97(35)};
#      [OID = 1.3.6.1.5.5.7.1.3]
```

```
#
#       Qualified Electronic Certificate Type Statement: QSIGN, QWAC, QSEAL
#       -----------------------------------------------------------------
#       0.4.0.1862.1.6, QcType
#       0.4.0.1862.1.6.1, esign
#       0.4.0.1862.1.6.2, eseal
#       0.4.0.1862.1.6.3, web
#
#
#       PSD2 Qualified Statement
#       ------------------------
#       NOTE:
#       PSP can be authorized by its national competent authority (NCA) to act in one or more PSD2 roles
#
#       etsi-psd2-qcStatement QC-STATEMENT ::= {SYNTAX PSD2QcType IDENTIFIED BY id-etsi-psd2-qcStatement }
#       id-etsi-psd2-qcStatement OBJECT IDENTIFIER ::=
#       {
#         itu-t(0) identified-organization(4) etsi(0) psd2(19495) qcstatement(2) }
#         [OID = 0.4.0.19495.2]
#         PSD2QcType ::= SEQUENCE {rolesOfPSP RolesOfPSP, nCAName NCAName, nCAId NCAId}
#       }
#
#       The NCAName shall be plain text name in English provided by the NCA itself for purpose
#       of identification in certificates
#       NCAName ::= UTF8String (SIZE (1..256))
#
#       The NCAId shall contain information using the following structure in the presented order:
#            2 character ISO 3166-1 [8] country code representing the NCA country;
#            hyphen-minus "-" (0x2D (ASCII), U+002D (UTF-8)); and
#            2-8 character NCA identifier without country code (A-Z uppercase only, no separator).
#       NCAId ::= UTF8String (SIZE (1..256))
#
#       RolesOfPSP ::= SEQUENCE OF RoleOfPSP
#
#         RoleOfPSP ::= SEQUENCE{ roleOfPspOid RoleOfPspOid,roleOfPspName RoleOfPspName}
#
#          RoleOfPspOid ::= OBJECT IDENTIFIER
#          -- Object Identifier arc for roles of payment service providers defined in the present document
#
#          etsi-psd2-roles OBJECT IDENTIFIER ::=
#
```

```
#          { itu-t(0) identified-organization(4) etsi(0) psd2(19495) id-roles(1) }

#          [OID = 0.4.0.19495.1]

#

#          -- Account Servicing Payment Service Provider (PSP_AS) role

#          [OID = 0.4.0.19495.1.1]

#          id-psd2-role-psp-as OBJECT IDENTIFIER ::=

#          { itu-t(0) identified-organization(4) etsi(0) psd2(19495) id-roles(1) 1 }

#

#          -- Payment Initiation Service Provider (PSP_PI) role

#          [OID = 0.4.0.19495.1.2]

#          id-psd2-role-psp-pi OBJECT IDENTIFIER ::=

#          { itu-t(0) identified-organization(4) etsi(0) psd2(19495) id-roles(1) 2 }

#

#          -- Account Information Service Provider (PSP_AI) role

#          [OID = 0.4.0.19495.1.3]

#          id-psd2-role-psp-ai OBJECT IDENTIFIER ::=

#          { itu-t(0) identified-organization(4) etsi(0) psd2(19495) id-roles(1) 3 }

#

#          -- Payment Service Provider issuing card-based payment instruments (PSP_IC) role

#          [OID = 0.4.0.19495.1.4]

#          id-psd2-role-psp-ic OBJECT IDENTIFIER ::=

#          { itu-t(0) identified-organization(4) etsi(0) psd2(19495) id-roles(1) 4 }

#

#          -- Payment Service Provider role name corresponding with OID

#             (one of string: "PSP_AS", "PSP_PI ", "PSP_AI ", "PSP_IC ")

#          RoleOfPspName ::= UTF8String (SIZE(1..256))

#

# QCStatement DER encoded of above as a MINIMUM encoding for a valid QWAC, QSEAL or QSIG issued certificate

# Note:

# The below DER encoding may optionally contain extra QCStatements as defined by ETSI

# such as QcCompliance, QcLimitValue etc. These are outside of scope of this configuration.

#

# The DER encoding in hex format may be generated using an ASN1 editor. For example:

# See https://www.sysadmins.lv/projects/asn1editor/default.aspx

#

qcStatements=DER:3081813013060604008e4601063009060704008e46010603306a0606040081982702306030393011060704008198
2701020c065053505f504930110607040081982701030c065053505f414930110607040081982701040c065053505f49430c1b46696e6
16e6369616c20436f6e647563742041757468f726974790c0647422d464341
```

3.3 PSP_PI ROLE

The PSD2 statement can contain a single PSD2 role as shown below for role PSP_PI.



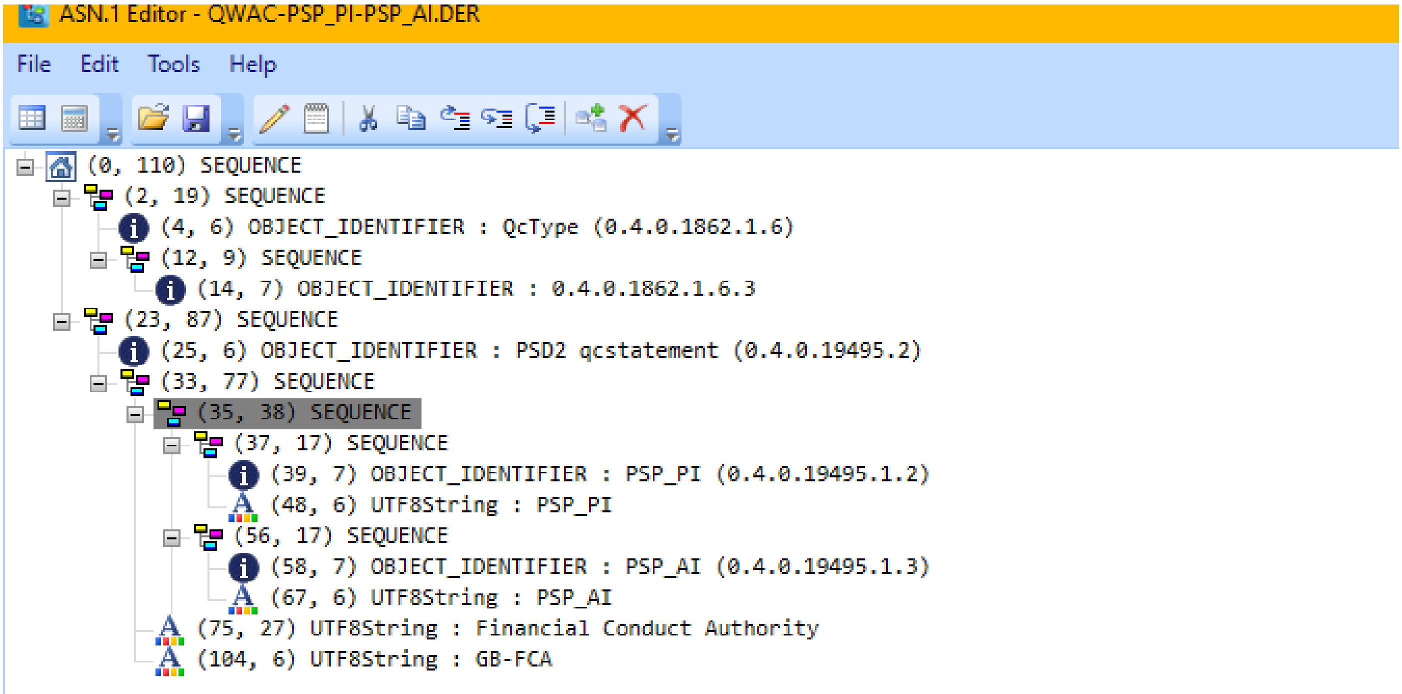
```
(0, 91) SEQUENCE
  (2, 19) SEQUENCE
    (4, 6) OBJECT_IDENTIFIER : QcType (0.4.0.1862.1.6)
    (12, 9) SEQUENCE
      (14, 7) OBJECT_IDENTIFIER : 0.4.0.1862.1.6.3
  (23, 68) SEQUENCE
    (25, 6) OBJECT_IDENTIFIER : PSD2 qcstatement (0.4.0.19495.2)
    (33, 58) SEQUENCE
      (35, 19) SEQUENCE
        (37, 17) SEQUENCE
          (39, 7) OBJECT_IDENTIFIER : PSP_PI (0.4.0.19495.1.2)
          (48, 6) UTF8String : PSP_PI
      (56, 27) UTF8String : Financial Conduct Authority
      (85, 6) UTF8String : GB-FCA
```

The OpensSSL der encoded QcStatement is shown below which can be used with the OpenSSL obwac.cnf file to generate a CSR for Open Banking PKI to sign

qcStatements=DER:305b3013060604008e4601063009060704008e4601060330440606040081982702303a30133011
0607040081982701020c065053505f50490c1b46696e616e6369616c20436f6e6475637420417574686f726974790c0
647422d464341

3.4 PSP_PI AND PSP_AI ROLES

The PSD2 statement can contain a list of PSD2 roles. An example is shown below for roles PSP_PI and PSP_AI.



```
(0, 110) SEQUENCE
  (2, 19) SEQUENCE
    (4, 6) OBJECT_IDENTIFIER : QcType (0.4.0.1862.1.6)
    (12, 9) SEQUENCE
      (14, 7) OBJECT_IDENTIFIER : 0.4.0.1862.1.6.3
  (23, 87) SEQUENCE
    (25, 6) OBJECT_IDENTIFIER : PSD2 qcstatement (0.4.0.19495.2)
    (33, 77) SEQUENCE
      (35, 38) SEQUENCE
        (37, 17) SEQUENCE
          (39, 7) OBJECT_IDENTIFIER : PSP_PI (0.4.0.19495.1.2)
          (48, 6) UTF8String : PSP_PI
        (56, 17) SEQUENCE
          (58, 7) OBJECT_IDENTIFIER : PSP_AI (0.4.0.19495.1.3)
          (67, 6) UTF8String : PSP_AI
      (75, 27) UTF8String : Financial Conduct Authority
      (104, 6) UTF8String : GB-FCA
```

The OpenSSL der encoded QcStatement is shown below which can be used with the OpenSSL obwac.cnf file to generate a CSR for Open Banking PKI to sign

qcStatements=DER:306e3013060604008e4601063009060704008e46010603305706060400819827023 04d30263011
060704008198270102 0c065053505f504930110607040081982701030c065053505f41490c1b46696e616e6369616c2
0436f6e6475637420417574686f726974790c0647422d464341

3.5 COMBINATIONS OF ALL PSP_XX ROLES: PSP_AS + PSP_PI + PSP_AI + PSP_IC

PSP_AS

qcStatements=DER:305b3013060604008e4601063009060704008e46010603304406060400819827023 03a30133011
0607040081982701010c065053505f41530c1b46696e616e6369616c20436f6e6475637420417574686f726974790c0
647422d464341

PSP_PI

qcStatements=DER:305b3013060604008e4601063009060704008e46010603304406060400819827023 03a30133011
0607040081982701020c065053505f50490c1b46696e616e6369616c20436f6e6475637420417574686f726974790c0
647422d464341

PSP_AI

qcStatements=DER:305b3013060604008e4601063009060704008e46010603304406060400819827023 03a30133011
0607040081982701030c065053505f41490c1b46696e616e6369616c20436f6e6475637420417574686f726974790c0
647422d464341

PSP_IC

qcStatements=DER:305b3013060604008e4601063009060704008e46010603304406060400819827023 03a30133011
0607040081982701040c065053505f49430c1b46696e616e6369616c20436f6e6475637420417574686f726974790c0
647422d464341

PSP_AS PSP_PI

qcStatements=DER:306e3013060604008e4601063009060704008e46010603305706060400819827023 04d30263011
0607040081982701010c065053505f415330110607040081982701020c065053505f50490c1b46696e616e6369616c2
0436f6e6475637420417574686f726974790c0647422d464341

PSP_AS PSP_AI

qcStatements=DER:306e3013060604008e4601063009060704008e46010603305706060400819827023 04d30263011
0607040081982701010c065053505f415330110607040081982701030c065053505f41490c1b46696e616e6369616c2
0436f6e6475637420417574686f726974790c0647422d464341

PSP_AS PSP_IC

qcStatements=DER:306e3013060604008e4601063009060704008e46010603305706060400819827023 04d30263011
0607040081982701010c065053505f415330110607040081982701040c065053505f49430c1b46696e616e6369616c2
0436f6e6475637420417574686f726974790c0647422d464341

PSP_PI PSP_AI

qcStatements=DER:306e3013060604008e4601063009060704008e46010603305706060400819827023 04d30263011
0607040081982701020c065053505f504930110607040081982701030c065053505f41490c1b46696e616e6369616c2
0436f6e6475637420417574686f726974790c0647422d464341

PSP_PI PSP_IC

qcStatements=DER:306e3013060604008e460106300906070400 8e460106033057060604008198270230 4d30263011
060704008198270102 0c065053505f5049301106070400819827010 40c065053505f49430c1b46696e616e6369616c2
0436f6e647563742041757468 6f726974790c0647422d464341

PSP_AI PSP_IC

qcStatements=DER:306e3013060604008e46010630090607 04008e46010603305706060400819827 02304d30263011
06070400819827010 30c065053505f4149301106070400819827010 40c065053505f49430c1b46696e616e6369616c2
0436f6e6475637420417574686f726974790c0647422d464341

PSP_AS PSP_PI PSP_AI

qcStatements=DER:3081813013060604008e460106300906070400 8e46010603306a0606040081982702306030393 0
110607040081982701010c065053505f415330110607040081982701020c065053505f504930110607040081982701
030c065053505f41490c1b46696e616e6369616c20436f6e6475637420417574686f726974790c0647422d464341

PSP_AS PSP_PI PSP_IC

qcStatements=DER:3081813013060604008e4601063009060704008e46010603306a060604008198270230603039 30
110607040081982701010c065053505f415330110607040081982701020c065053505f504930110607040081982701
040c065053505f49430c1b46696e616e6369616c20436f6e6475637420417574686f726974790c0647422d464341

PSP_AS PSP_AI PSP_IC

qcStatements=DER:3081813013060604008e4601063009060704008e46010603306a06060400819827023060303930
110607040081982701010c065053505f415330110607040081982701030c065053505f414930110607040081982701
040c065053505f49430c1b46696e616e6369616c20436f6e6475637420417574686f726974790c0647422d464341

PSP_PI PSP_AI PSP_IC

qcStatements=DER:3081813013060604008e4601063009060704008e46010603306a06060400819827023060303930
110607040081982701020c065053505f504930110607040081982701030c065053505f414930110607040081982701
040c065053505f49430c1b46696e616e6369616c20436f6e6475637420417574686f726974790c0647422d464341

PSP_AS PSP_PI PSP_AI PSP_IC

qcStatements=DER:3081943013060604008e4601063009060704008e46010603307d0606040081982702307330 4c30
110607040081982701010c065053505f415330110607040081982701020c065053505f504930110607040081982701
030c065053505f414930110607040081982701040c065053505f49430c1b46696e616e6369616c20436f6e647563742
0417574686f726974790c0647422d464341

3.6 EXAMPLE OBWAC CSR CREATION FOR ROLES PSP_PI, PSP_AI AND PSP_IC

```
# vi obwac.cnf
```

```
<replace qcStatements=DER:xxxx with qcStatements listed above for PSP_PI PSP_AI PSP_IC>

<save file>
```

```
# openssl req -new -config obwac.cnf -out obwac.csr -keyout obwac.key

# cat obwac.csr
```

```
-----BEGIN CERTIFICATE REQUEST-----

MIIDvzCCAqcCAQAwgYIxCzAJBgNVBAYTAkdCMSEwHwYDVQQKDBhPcGVuIEJhbmtp

bmcgTGltaXRlZCAoRCkxLzAtBgNVBGEMJlBTREdCLU9CLVVua25vd25pOWtzSHNI

Vndtc0xlbGpPTTJjZTdQMR8wHQYDVQQDDBZpOWtzSHNIVndtc0xlbGpPTTJjZTdQ
```

```
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAw1BsSBrs0ZlfqvDUurD1
vdrLqKE4asPnFS6cWw2fwzCnmKhXpu4HkyVYpwlZX3a2yDg8LrelVjPyn+U+bY43
+N67wQwOTgVPBrueMDRDgDJOEUNNBror6NQcnP5KnpShGyzACKZELLYMzByBY7AM
AXF00nC8RnJzoI46NdgiFjEVj4RIqa3EI2RTbLgzun3mIlDXcmwPbUwe3BSQjU2j
ctTIru8hM3gX2VFMqHGwChnuTWtvINTO8dCWwT+qEEEFLas+7aqDzqd5X4bQOVeX
d1asEF1ELOUGH3Hxn8TgvSVKKwx1xv9/R9dT5APFQFJIC+VuxGUYd5PS9jDewvbc
hQIDAQABoIH2MIHzBgkqhkiG9w0BCQ4xgeUwgeIwDgYDVR0PAQH/BAQDAgeAMB0G
A1UdJQQWMBQGCCsGAQUFBwMCBggrBgEFBQcDATAdBgNVHQ4EFgQU46uuLJ3ieyyS
t4rdy7i9fWEbnQ0wgZEGCCsGAQUFBwEDBIGEMIGBMBMGBgQAjkYBBjAJBgcEAI5G
AQYDMGoGBgQAgZgnAjBgMDkwEQYHBACBmCcBAgwGUFNQX1BJMBEGBwQAgZgnAQMM
BlBTUF9BSTARBgcEAIGYJwEEDAZQU1BfSUMMG0ZpbmFuY2lhbCBDb25kdWN0IEF1
dGhvcml0eQwGR0ItRkNBMA0GCSqGSIb3DQEBCwUAA4IBAQAbnIpKjzKnxdT06wUg
7h7ghkj4rKL3JMNugwT+W7TtBSbPtHIlZLHK8gBnimTzQIt736E1Aw0vybYwCFE7
9IRMveGkjj0J7W3fzHcgXczlMchEsIIzsFSHDNyBNrJuszlj9mtPtagQjkzux9E6
pW7YCqKXP8lrsBnydVFOSXAAglax2xqYk0ssyFnA/pIskzUAMKcgneXOyPTLO89T
HSMEP18jswB0XWhWvMAr+FRjDkb7TNUL6WniXr5+1oXb+21ZUmBp/7zyy5fGssxo
2QBEiHBELDQMz7Oiuvgtk/Fr3UJX6JQXdFYUNo3lJy5qk4TRUXk5f2bJpDLGI+sc
tKAK
-----END CERTIFICATE REQUEST-----
```

```
# openssl asn1parse -in obwac.csr -inform PEM
0:d=0  hl=4 l= 959 cons: SEQUENCE
4:d=1  hl=4 l= 679 cons: SEQUENCE
8:d=2  hl=2 l=   1 prim: INTEGER           :00
11:d=2  hl=3 l= 130 cons: SEQUENCE
14:d=3  hl=2 l=  11 cons: SET
16:d=4  hl=2 l=   9 cons: SEQUENCE
18:d=5  hl=2 l=   3 prim: OBJECT            :countryName
23:d=5  hl=2 l=   2 prim: PRINTABLESTRING   :GB
27:d=3  hl=2 l=  33 cons: SET
29:d=4  hl=2 l=  31 cons: SEQUENCE
31:d=5  hl=2 l=   3 prim: OBJECT            :organizationName
36:d=5  hl=2 l=  24 prim: UTF8STRING        :Open Banking Limited (D)
62:d=3  hl=2 l=  47 cons: SET
64:d=4  hl=2 l=  45 cons: SEQUENCE
66:d=5  hl=2 l=   3 prim: OBJECT            :2.5.4.97
71:d=5  hl=2 l=  38 prim: UTF8STRING        :PSDGB-OB-Unknowni9ksHsHVwmsLeljOM2ce7P
111:d=3  hl=2 l=  31 cons: SET
113:d=4  hl=2 l=  29 cons: SEQUENCE
115:d=5  hl=2 l=   3 prim: OBJECT            :commonName
120:d=5  hl=2 l=  22 prim: UTF8STRING        :i9ksHsHVwmsLeljOM2ce7P
```

```
144:d=2  hl=4 l= 290 cons: SEQUENCE
148:d=3  hl=2 l=  13 cons: SEQUENCE
150:d=4  hl=2 l=   9 prim: OBJECT            :rsaEncryption
161:d=4  hl=2 l=   0 prim: NULL
163:d=3  hl=4 l= 271 prim: BIT STRING
438:d=2  hl=3 l= 246 cons: cont [ 0 ]
441:d=3  hl=3 l= 243 cons: SEQUENCE
444:d=4  hl=2 l=   9 prim: OBJECT            :Extension Request
455:d=4  hl=3 l= 229 cons: SET
458:d=5  hl=3 l= 226 cons: SEQUENCE
461:d=6  hl=2 l=  14 cons: SEQUENCE
463:d=7  hl=2 l=   3 prim: OBJECT            :X509v3 Key Usage
468:d=7  hl=2 l=   1 prim: BOOLEAN           :255
471:d=7  hl=2 l=   4 prim: OCTET STRING      [HEX DUMP]:03020780
477:d=6  hl=2 l=  29 cons: SEQUENCE
479:d=7  hl=2 l=   3 prim: OBJECT            :X509v3 Extended Key Usage
484:d=7  hl=2 l=  22 prim: OCTET STRING      [HEX DUMP]:301406082B0601050507030206082B06010505070301
508:d=6  hl=2 l=  29 cons: SEQUENCE
510:d=7  hl=2 l=   3 prim: OBJECT            :X509v3 Subject Key Identifier
515:d=7  hl=2 l=  22 prim: OCTET STRING      [HEX DUMP]:0414E3ABAE2C9DE27B2C92B78ADDCBB8BD7D611B9D0D
539:d=6  hl=3 l= 145 cons: SEQUENCE
542:d=7  hl=2 l=   8 prim: OBJECT            :qcStatements
552:d=7  hl=3 l= 132 prim: OCTET STRING      [HEX
        DUMP]:308181301306060604008E4601063009060704008E46010603306A06060400819827023060303930110607040081982701020C065053505F504930110607040081982701030C065053505F414930110607040081982701040C065053505F49430C1B46696E616E6369616C20436F6E64756374204175746686F726974790C0647422D464341
687:d=1  hl=2 l=  13 cons: SEQUENCE
689:d=2  hl=2 l=   9 prim: OBJECT            :sha256WithRSAEncryption
700:d=2  hl=2 l=   0 prim: NULL
702:d=1  hl=4 l= 257 prim: BIT STRING
```

The CSR may be uploaded to Open Banking PKI to issue an OBWAC public certificate associated with your private key.

4 OBSEAL X509 CERTIFICATE SIGNING REQUEST PROFILE

4.1 INTRODUCTION

This chapter demonstrates how to generate a valid CSR by a client that requires an OBSEAL x509 certificate issued to them.

4.2 OBSEAL OPENSSL CNF FILE

The following configuration file is used to generate a CSR for an OBSEAL certificate using OpenSSL

All text highlighted in red must be changed by the client to specific data pertinent to them.

NB.

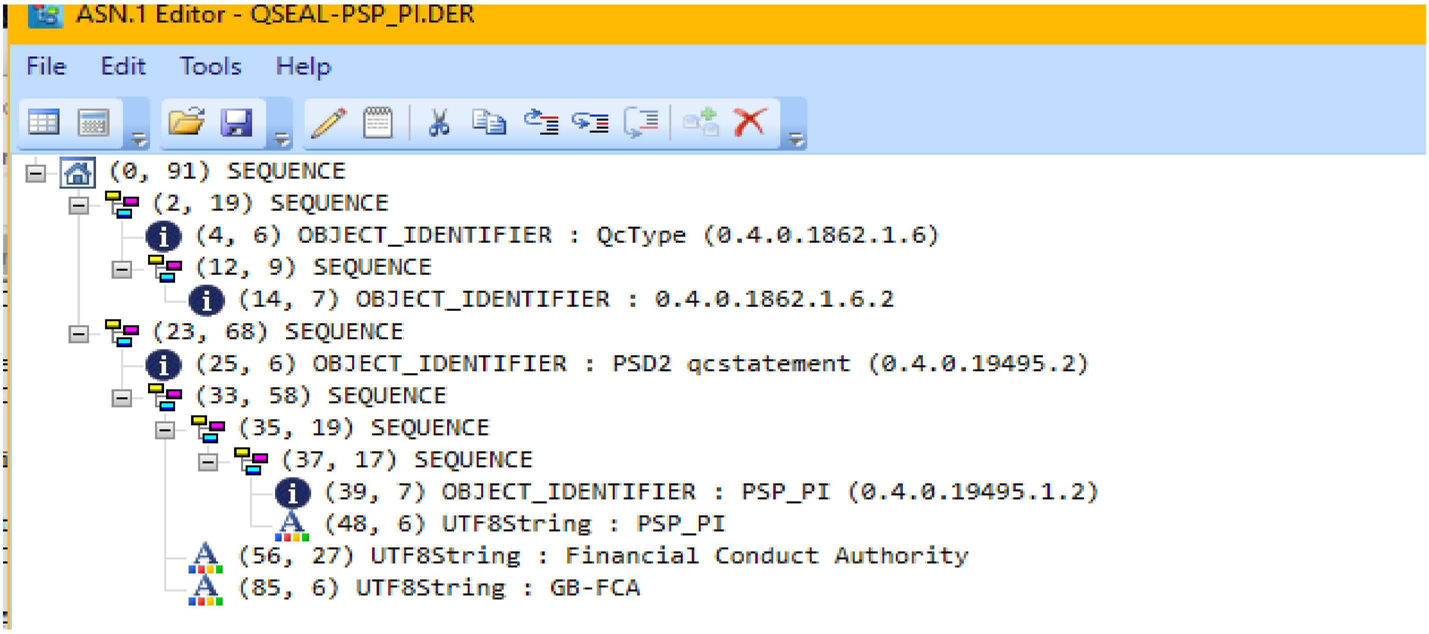
A client can use the OpenSSL cnf file below to generate a CSR which is subsequently signed using their own certificate authority. However, such a certificate will not be trusted by Open Banking.

obseal.cnf file for OpenSSL CSR request

```
#
# OPENSSL CSR REQUEST CONFIGURATION FILE
# ======================================
#
# OBSEAL qualified client certificate request with PSD2 role: PSP_PI PSP_AI
# -----------------------------------------------------------------------
# See latest specification: ETSI TS 119 495 V1.2.1 (2018-11)
# https://www.etsi.org/deliver/etsi_ts/119400_119499/119495/01.02.01_60/ts_119495v010201p.pdf
#
oid_section            = new_oids

[ new_oids ]
organizationIdentifier  = 2.5.4.97            # OpenSSL may not recognize this OID so need to add.

[ req ]
default_bits            = 2048                  # RSA key size
encrypt_key             = yes                   # Protect private key: yes or no. yes recommended
default_md              = sha256                # MD to use. sha256 recommended
utf8                    = yes                   # Input is UTF-8.
string_mask             = utf8only              # Emit UTF-8 strings
prompt                  = no                    # Prompt for DN. yes or no.
distinguished_name      = client_dn       # DN template. Mandatory to include organizationIdentifier
req_extensions          = client_reqext         # Desired extensions. Mandatory to include PSD2 QCStatements

#
# Subject Distinguished Name format in certificate
# -----------------------------------------------
# EG: CN = 0015800001041ReAAI, 2.5.4.97 = PSDGB-OB-Unknown0015800001041ReAAI, O = Open Banking Limited (D), C
= GB
#
#
[ client_dn ]
countryName             = "GB"                          # Country code - see doc above
organizationName        = "Open Banking Limited (D)"    # Organizational name
#
#  organizationIdentifier
#  ----------------------
#  The organizationIdentifier shall be present in the Subject's Distinguished Name
#  and encoded with legal person syntax
#
#  EXAMPLE: The organizationIdentifier "PSDPL-PFSA-1234567890" means a certificate issued to a PSP where
#  the authorization number is 1234567890, authorization was granted by the Polish Financial
#  Supervision Authority (identifier after second hyphen-minus is decided by Polish numbering
#  system). Other examples can include use of non-alphanumeric characters such as "PSDBE-NBB-
#  1234.567.890" and "PSDFI-FINFSA-1234567-8" and "PSDMT-MFSA-A 12345" (note space character after "A")
#
organizationIdentifier  = "PSDGB-OB-Unknown0015800001041ReAAI" # Must be in format as shown above
commonName              = "0015800001041ReAAI"                # Subject common name
#
# Required specific extensions in certificate
#
[ client_reqext ]
keyUsage                = critical,digitalSignature,nonRepudiation  # Must be critical
#
#
# extendedKeyUsage        = critical, document-signing               # Not used
subjectKeyIdentifier    = hash                                       # Hash value to calculate SKI
#
#
#   QC-STATEMENT
#
#      FROM PKIXqualified97 {iso(1) identified-organization(3) dod(6)
#      internet(1) security(5) mechanisms(5) pkix(7) id-mod(0) id-mod-qualified-cert-97(35)};
#      [OID = 1.3.6.1.5.5.7.1.3]
#
#       Qualified Electronic Certificate Type Statement: QSIGN, QWAC, QSEAL
#       -------------------------------------------------------------------
#       0.4.0.1862.1.6, QcType
#       0.4.0.1862.1.6.1, esign
#       0.4.0.1862.1.6.2, eseal
#       0.4.0.1862.1.6.3, web
#
#
#       PSD2 Qualified Statement
#       ------------------------
#       NOTE:
#       PSP can be authorized by its national competent authority (NCA) to act in one or more PSD2 roles
#
#       etsi-psd2-qcStatement QC-STATEMENT ::= {SYNTAX PSD2QcType IDENTIFIED BY id-etsi-psd2-qcStatement }
#       id-etsi-psd2-qcStatement OBJECT IDENTIFIER ::=
#       {
#         itu-t(0) identified-organization(4) etsi(0) psd2(19495) qcstatement(2) }
#         [OID = 0.4.0.19495.2]
#         PSD2QcType ::= SEQUENCE {rolesOfPSP RolesOfPSP, nCAName NCAName, nCAId NCAId}
#       }
#
#       The NCAName shall be plain text name in English provided by the NCA itself for purpose
#       of identification in certificates
#       NCAName ::= UTF8String (SIZE (1..256))
#
#
#       The NCAId shall contain information using the following structure in the presented order:
#            2 character ISO 3166-1 [8] country code representing the NCA country;
#            hyphen-minus "-" (0x2D (ASCII), U+002D (UTF-8)); and
#            2-8 character NCA identifier without country code (A-Z uppercase only, no separator).
#       NCAId ::= UTF8String (SIZE (1..256))
#
#       RolesOfPSP ::= SEQUENCE OF RoleOfPSP
#
#         RoleOfPSP ::= SEQUENCE{ roleOfPspOid RoleOfPspOid,roleOfPspName RoleOfPspName}
#
#          RoleOfPspOid ::= OBJECT IDENTIFIER
#          -- Object Identifier arc for roles of payment service providers defined in the present document
#
#          etsi-psd2-roles OBJECT IDENTIFIER ::=
#
#          { itu-t(0) identified-organization(4) etsi(0) psd2(19495) id-roles(1) }
#          [OID = 0.4.0.19495.1]
#
#          -- Account Servicing Payment Service Provider (PSP_AS) role
#           [OID = 0.4.0.19495.1.1]
#           id-psd2-role-psp-as OBJECT IDENTIFIER ::=
#           { itu-t(0) identified-organization(4) etsi(0) psd2(19495) id-roles(1) 1 }
#
#           -- Payment Initiation Service Provider (PSP_PI) role
#           [OID = 0.4.0.19495.1.2]
#           id-psd2-role-psp-pi OBJECT IDENTIFIER ::=
#           { itu-t(0) identified-organization(4) etsi(0) psd2(19495) id-roles(1) 2 }
#
#           -- Account Information Service Provider (PSP_AI) role
#           [OID = 0.4.0.19495.1.3]
#           id-psd2-role-psp-ai OBJECT IDENTIFIER ::=
#           { itu-t(0) identified-organization(4) etsi(0) psd2(19495) id-roles(1) 3 }
#
#           -- Payment Service Provider issuing card-based payment instruments (PSP_IC) role
#           [OID = 0.4.0.19495.1.4]
#           id-psd2-role-psp-ic OBJECT IDENTIFIER ::=
#           { itu-t(0) identified-organization(4) etsi(0) psd2(19495) id-roles(1) 4 }
#
#           -- Payment Service Provider role name corresponding with OID
#              (one of string: "PSP_AS", "PSP_PI ", "PSP_AI ", "PSP_IC ")
#           RoleOfPspName ::= UTF8String (SIZE(1..256))
#
# QCStatement DER encoded of above as a MINIMUM encoding for a valid QWAC, QSEAL or QSIG issued certificate
# Note:
# The below DER encoding may optionally contain extra QCStatements as defined by ETSI
# such as QcCompliance, QcLimitValue etc. These are outside of scope of this configuration.
#
# The DER encoding in hex format may be generated using an ASN1 editor. For example:
# See https://www.sysadmins.lv/projects/asn1editor/default.aspx
#
qcStatements=DER:306e3013060604008e46010630090607040008e46010602305706060400819827020304d30263011060704008198270
1020c065053505f504930110607040081982701030c065053505f41490c1b46696e616e6369616c20436f6e647563742041757468 6f7
26974790c0647422d464341
```

4.3 PSP_PI ROLE

The PSD2 statement can contain a single PSD2 role as shown below for role PSP_PI.



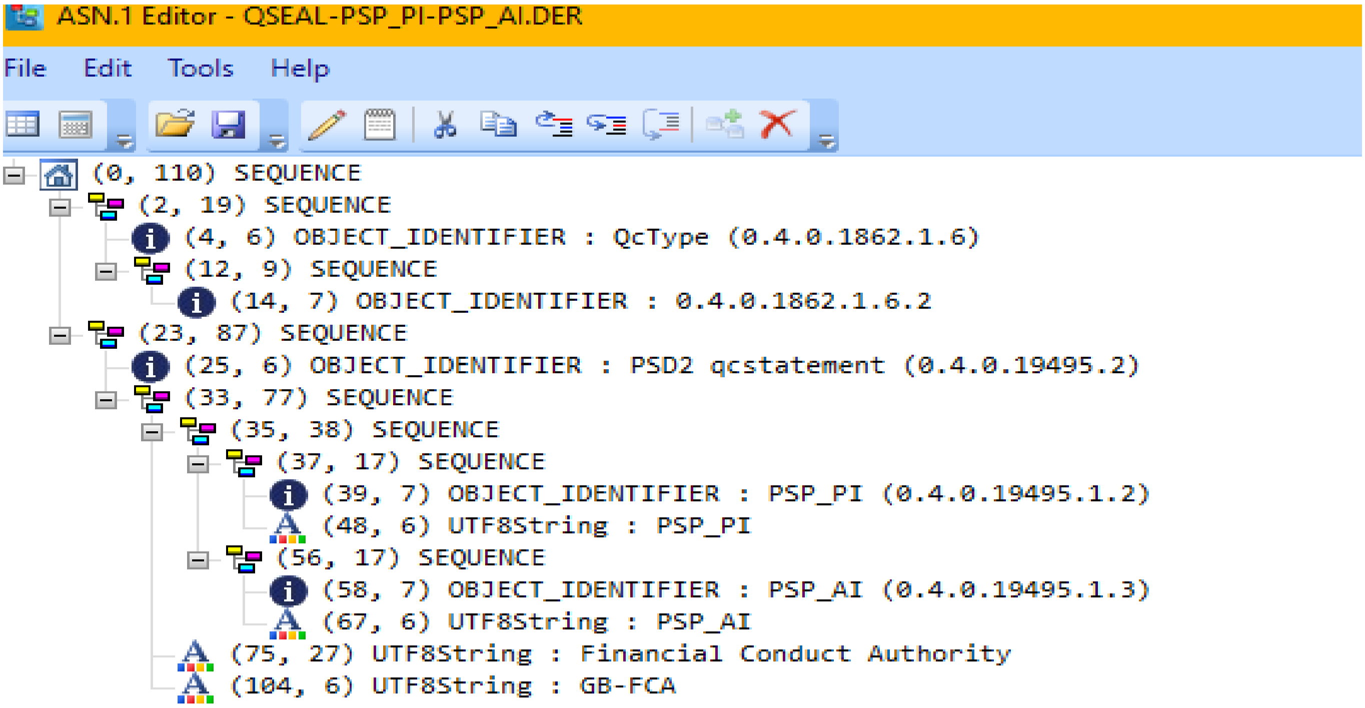
```
(0, 91) SEQUENCE
  (2, 19) SEQUENCE
    (4, 6) OBJECT_IDENTIFIER : QcType (0.4.0.1862.1.6)
    (12, 9) SEQUENCE
      (14, 7) OBJECT_IDENTIFIER : 0.4.0.1862.1.6.2
  (23, 68) SEQUENCE
    (25, 6) OBJECT_IDENTIFIER : PSD2 qcstatement (0.4.0.19495.2)
    (33, 58) SEQUENCE
      (35, 19) SEQUENCE
        (37, 17) SEQUENCE
          (39, 7) OBJECT_IDENTIFIER : PSP_PI (0.4.0.19495.1.2)
          (48, 6) UTF8String : PSP_PI
      (56, 27) UTF8String : Financial Conduct Authority
      (85, 6) UTF8String : GB-FCA
```

The OpensSSL der encoded QcStatement is shown below which can be used with the OpenSSL obseal.cnf file to generate a CSR for Open Banking PKI to sign

qcStatements=DER:305b3013060604008e4601063009060704008e4601060230440606040081982702303a30133011 0607040081982701020c065053505f50490c1b46696e616e6369616c20436f6e647563742041757468f726974790c0 647422d464341

4.4 PSP_PI AND PSP_AI ROLES

The PSD2 statement can contain a list of PSD2 roles. An example is shown below for roles PSP_PI and PSP_AI.



```
(0, 110) SEQUENCE
  (2, 19) SEQUENCE
    (4, 6) OBJECT_IDENTIFIER : QcType (0.4.0.1862.1.6)
    (12, 9) SEQUENCE
      (14, 7) OBJECT_IDENTIFIER : 0.4.0.1862.1.6.2
  (23, 87) SEQUENCE
    (25, 6) OBJECT_IDENTIFIER : PSD2 qcstatement (0.4.0.19495.2)
    (33, 77) SEQUENCE
      (35, 38) SEQUENCE
        (37, 17) SEQUENCE
          (39, 7) OBJECT_IDENTIFIER : PSP_PI (0.4.0.19495.1.2)
          (48, 6) UTF8String : PSP_PI
        (56, 17) SEQUENCE
          (58, 7) OBJECT_IDENTIFIER : PSP_AI (0.4.0.19495.1.3)
          (67, 6) UTF8String : PSP_AI
      (75, 27) UTF8String : Financial Conduct Authority
      (104, 6) UTF8String : GB-FCA
```

The OpenSSL der encoded QcStatement is shown below which can be used with the OpenSSL obseal.cnf file to generate a CSR for Open Banking PKI to sign

```
qcStatements=DER:306e3013060604008e4601063009060704008e46010602305706060400819827 02304d302630110607040081982701020c065053505f50493011060704008198270103 0c065053505f41490c1b46696e616e6369616c20436f6e647563742041757468 6f726974790c0647422d464341
```

4.5 COMBINATIONS OF ALL PSP_XX ROLES: PSP_AS + PSP_PI + PSP_AI + PSP_IC

PSP_AS

```
qcStatements=DER:305b3013060604008e4601063009060704008e460106023044060604008198270 2303a301330110607040081982701010c065053505f41530c1b46696e616e6369616c2 0436f6e6475637420417574686f726974790c0647422d464341
```

PSP_PI

```
qcStatements=DER:305b3013060604008e4601063009060704008e460106023044060604008198270 2303a301330110607040081982701020c065053505f50490c1b46696e616e6369616c2 0436f6e6475637420417574686f726974790c0647422d464341
```

PSP_AI

```
qcStatements=DER:305b3013060604008e4601063009060704008e460106023044060604008198270 2303a301330110607040081982701030c065053505f41490c1b46696e616e6369616c2 0436f6e6475637420417574686f726974790c0647422d464341
```

PSP_IC

```
qcStatements=DER:305b3013060604008e4601063009060704008e460106023044060604008198270 2303a301330110607040081982701040c065053505f49430c1b46696e616e6369616c2 0436f6e6475637420417574686f726974790c0647422d464341
```

PSP_AS PSP_PI

```
qcStatements=DER:306e3013060604008e4601063009060704008e460106023057060604008198270 2304d302630110607040081982701010c065053505f4153301106070400819827010 20c065053505f50490c1b46696e616e6369616c20436f6e6475637420417574686f72 6974790c0647422d464341
```

PSP_AS PSP_AI

```
qcStatements=DER:306e3013060604008e4601063009060704008e460106023057060604008198270 2304d302630110607040081982701010c065053505f4153301106070400819827010 30c065053505f41490c1b46696e616e6369616c20436f6e6475637420417574686f72 6974790c0647422d464341
```

PSP_AS PSP_IC

```
qcStatements=DER:306e3013060604008e4601063009060704008e460106023057060604008198270 2304d302630110607040081982701010c065053505f4153301106070400819827010 40c065053505f49430c1b46696e616e6369616c20436f6e6475637420417574686f72 6974790c0647422d464341
```

PSP_PI PSP_AI

```
qcStatements=DER:306e3013060604008e4601063009060704008e460106023057060604008198270 2304d302630110607040081982701020c065053505f5049301106070400819827010 30c065053505f41490c1b46696e616e6369616c20436f6e6475637420417574686f72 6974790c0647422d464341
```

PSP_PI PSP_IC

```
qcStatements=DER:306e3013060604008e4601063009060704008e46010602305706060400819827023304d30263011
060704008198270102 0c065053505f5049301106070400819827010 40c065053505f49430c1b46696e616e6369616c2
0436f6e647563742041757468f726974790c0647422d464341
```

PSP_AI PSP_IC

```
qcStatements=DER:306e3013060604008e4601063009060704008e46010602305706060400819827023304d30263011
0607040081982701030c065053505f414930110607040081982701040c065053505f49430c1b46696e616e6369616c2
0436f6e647563742041757468f726974790c0647422d464341
```

PSP_AS PSP_PI PSP_AI

```
qcStatements=DER:3081813013060604008e4601063009060704008e46010602306a0606040081982702306030393 0
110607040081982701010c065053505f415330110607040081982701020c065053505f504930110607040081982701
030c065053505f41490c1b46696e616e6369616c20436f6e647563742041757468f726974790c0647422d464341
```

PSP_AS PSP_PI PSP_IC

```
qcStatements=DER:3081813013060604008e4601063009060704008e46010602306a06060400819827023060303930
110607040081982701010c065053505f415330110607040081982701020c065053505f504930110607040081982701
040c065053505f49430c1b46696e616e6369616c20436f6e647563742041757468f726974790c0647422d464341
```

PSP_AS PSP_AI PSP_IC

```
qcStatements=DER:3081813013060604008e4601063009060704008e46010602306a06060400819827023060303930
110607040081982701010c065053505f415330110607040081982701030c065053505f414930110607040081982701
040c065053505f49430c1b46696e616e6369616c20436f6e647563742041757468f726974790c0647422d464341
```

PSP_PI PSP_AI PSP_IC

```
qcStatements=DER:3081813013060604008e4601063009060704008e46010602306a06060400819827023060303930
110607040081982701020c065053505f504930110607040081982701030c065053505f414930110607040081982701
040c065053505f49430c1b46696e616e6369616c20436f6e647563742041757468f726974790c0647422d464341
```

PSP_AS PSP_PI PSP_AI PSP_IC

```
qcStatements=DER:3081943013060604008e4601063009060704008e46010602307d06060400819827023073304c30
110607040081982701010c065053505f415330110607040081982701020c065053505f504930110607040081982701
030c065053505f414930110607040081982701040c065053505f49430c1b46696e616e6369616c20436f6e647563742
0417574686f726974790c0647422d464341
```

4.6 EXAMPLE OBSEAL CSR CREATION FOR ROLES PSP_AS AND PSP_IC

```
# vi obseal.cnf
```

```
<replace qcStatements=DER:xxxx with qcStatements listed above for PSP_AS PSP_IC>

<save file>
```

```
# openssl req -new -config obseal.cnf -out obseal.csr -keyout obseal.key
```

```
# cat obseal.csr
```

```
-----BEGIN CERTIFICATE REQUEST-----

MIIDgTCCAmkCAQAwejELMAkGA1UEBhMCR0IxITAfBgNVBAoMGE9wZW4gQmFua2lu

ZyBMaW1pdGVkIChEKTErMCkGA1UEYQwiUFNER0ItT0ItVW5rbm93bjAwMTU4MDAw

MDEwNDFSZUFBSTEbMBkGA1UEAwwSMDAxNTgwMDAwMTA0MVJlQUFJMIIBIjANBgkq
```

```
hkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAr+6WZcGI44s3ruf3OM9puCnSSPQYWiYU

UPnwYKNLhAgh0k57iw88eMuKdGPlYOv8ZaBwKnxWNlYLO+TgXU0uYgfYv1dY6fsG

A1CFwe9tKNNqaR2iavITQ1w31yn01TktmMNAhwtq6FV/zzEqQHHwn/acXTNPev+O

zFPcJPcOtPI7C24iFIMzEgG5KKeR9XrfdDGIaLKd4DsAov2CtGjgYp5139iKB3qX

A7WXINJ5dYhqD+2T7DNfx6SnNStriO6xsC6mb9kgT1Ie1JTrE6MFeW3DEH8MQ7ye

9MPJmSDXBVcHB5vk+9eeyCKSR61BhUqMZpF3p2p/98Qjy6TVcpYYowIDAQABoIHB

MIG+BgkqhkiG9w0BCQ4xgbAwga0wDgYDVR0PAQH/BAQDAgbAMB0GA1UdDgQWBBTo

relMTXOLrWsEMuu/CpP7S7IGyzB8BggrBgEFBQcBAwRwMG4wEwYGBACORgEGMAkG

BwQAjkYBBgIwVwYGBACBmCcCME0wJjARBgcEAIGYJwEBDAZQU1BfQVMwEQYHBACB

mCcBBAwGUFNQX0lDDBtGaW5hbmNpYWwgQ29uZHVjdCBBdXRob3JpdHkMBkdCLUZD

QTANBgkqhkiG9w0BAQsFAAOCAQEAr+m0RWH5kp9/NJr1ZtXc7/MSQ0tLCo+/0v6t

V/CAyySb2A1K1S3D4TslyIlcNqKbz5HkgpLgAKCX2RvIRLYNVAkS8hxf6B4Gf6jo

dTEfbRH6AZhtT4QaJ/MKq8e0upcNvd/eOnqmbS64JGuVp7Y7+UVeOT+CQmwX62uJ

sGJdBsVkT/81hTNNVNvru1ews0XZ8HlSohtzde9HfF4u72EJwBVcpCDrt/wjVocd

myKA9hrcauXE+9auupC9y569PQFxnqvTBXZiPUQbLWGgB0HfpEPvkOnVw7Uw1DDH

ISwFMFXlz3vH8l2hNAzp2CkU+Qa8FA0lw4W14wCPybdygDYvFA==

-----END CERTIFICATE REQUEST-----
```

# openssl asn1parse -in obseal.csr -inform PEM

```
0:d=0  hl=4 l= 897 cons: SEQUENCE

4:d=1  hl=4 l= 617 cons: SEQUENCE

8:d=2  hl=2 l=   1 prim: INTEGER           :00

11:d=2  hl=2 l= 122 cons: SEQUENCE

13:d=3  hl=2 l=  11 cons: SET

15:d=4  hl=2 l=   9 cons: SEQUENCE

17:d=5  hl=2 l=   3 prim: OBJECT            :countryName

22:d=5  hl=2 l=   2 prim: PRINTABLESTRING   :GB

26:d=3  hl=2 l=  33 cons: SET

28:d=4  hl=2 l=  31 cons: SEQUENCE

30:d=5  hl=2 l=   3 prim: OBJECT            :organizationName

35:d=5  hl=2 l=  24 prim: UTF8STRING        :Open Banking Limited (D)

61:d=3  hl=2 l=  43 cons: SET

63:d=4  hl=2 l=  41 cons: SEQUENCE

65:d=5  hl=2 l=   3 prim: OBJECT            :2.5.4.97

70:d=5  hl=2 l=  34 prim: UTF8STRING        :PSDGB-OB-Unknown0015800001041ReAAI

106:d=3  hl=2 l=  27 cons: SET

108:d=4  hl=2 l=  25 cons: SEQUENCE

110:d=5  hl=2 l=   3 prim: OBJECT            :commonName

115:d=5  hl=2 l=  18 prim: UTF8STRING        :0015800001041ReAAI

135:d=2  hl=4 l= 290 cons: SEQUENCE

139:d=3  hl=2 l=  13 cons: SEQUENCE

141:d=4  hl=2 l=   9 prim: OBJECT            :rsaEncryption
```

```
152:d=4  hl=2 l=   0 prim: NULL
154:d=3  hl=4 l= 271 prim: BIT STRING
429:d=2  hl=3 l= 193 cons: cont [ 0 ]
432:d=3  hl=3 l= 190 cons: SEQUENCE
435:d=4  hl=2 l=   9 prim: OBJECT            :Extension Request
446:d=4  hl=3 l= 176 cons: SET449:d=5  hl=3 l= 173 cons: SEQUENCE
452:d=6  hl=2 l=  14 cons: SEQUENCE
454:d=7  hl=2 l=   3 prim: OBJECT            :X509v3 Key Usage
459:d=7  hl=2 l=   1 prim: BOOLEAN           :255
462:d=7  hl=2 l=   4 prim: OCTET STRING      [HEX DUMP]:030206C0
468:d=6  hl=2 l=  29 cons: SEQUENCE
470:d=7  hl=2 l=   3 prim: OBJECT            :X509v3 Subject Key Identifier
475:d=7  hl=2 l=  22 prim: OCTET STRING      [HEX DUMP]:0414E8ADE94C4D738BAD6B0432EBBF0A93FB4BB206CB
499:d=6  hl=2 l= 124 cons: SEQUENCE
501:d=7  hl=2 l=   8 prim: OBJECT            :qcStatements
511:d=7          hl=2      l=      112      prim:      OCTET      STRING                                [HEX
DUMP]:306E301306060400 8E4601063009060704008E460106023057060604008198270230 4D302630110607040081982701010C06505
3505F41533011060704008198270104 0C065053505F49430C1B46696E616E6369616C20436F6E64756374204175746 86F726974790C06
47422D464341
625:d=1  hl=2 l=  13 cons: SEQUENCE
627:d=2  hl=2 l=   9 prim: OBJECT            :sha256WithRSAEncryption
638:d=2  hl=2 l=   0 prim: NULL
640:d=1  hl=4 l= 257 prim: BIT STRING
```

The CSR may be uploaded to Open Banking PKI to issue an OBSEAL public certificate associated with your private key.