



UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

MESTRADO EM ENGENHARIA INFORMÁTICA

CRİPTOGRAFIA E SEGURANÇA DE INFORMAÇÃO

Engenharia de Segurança

Ficha Prática 5

Grupo Nº 3

Ariana Lousada (PG47034) Luís Carneiro (PG46541)
Rui Cardoso (PG42849)

19 de abril de 2022

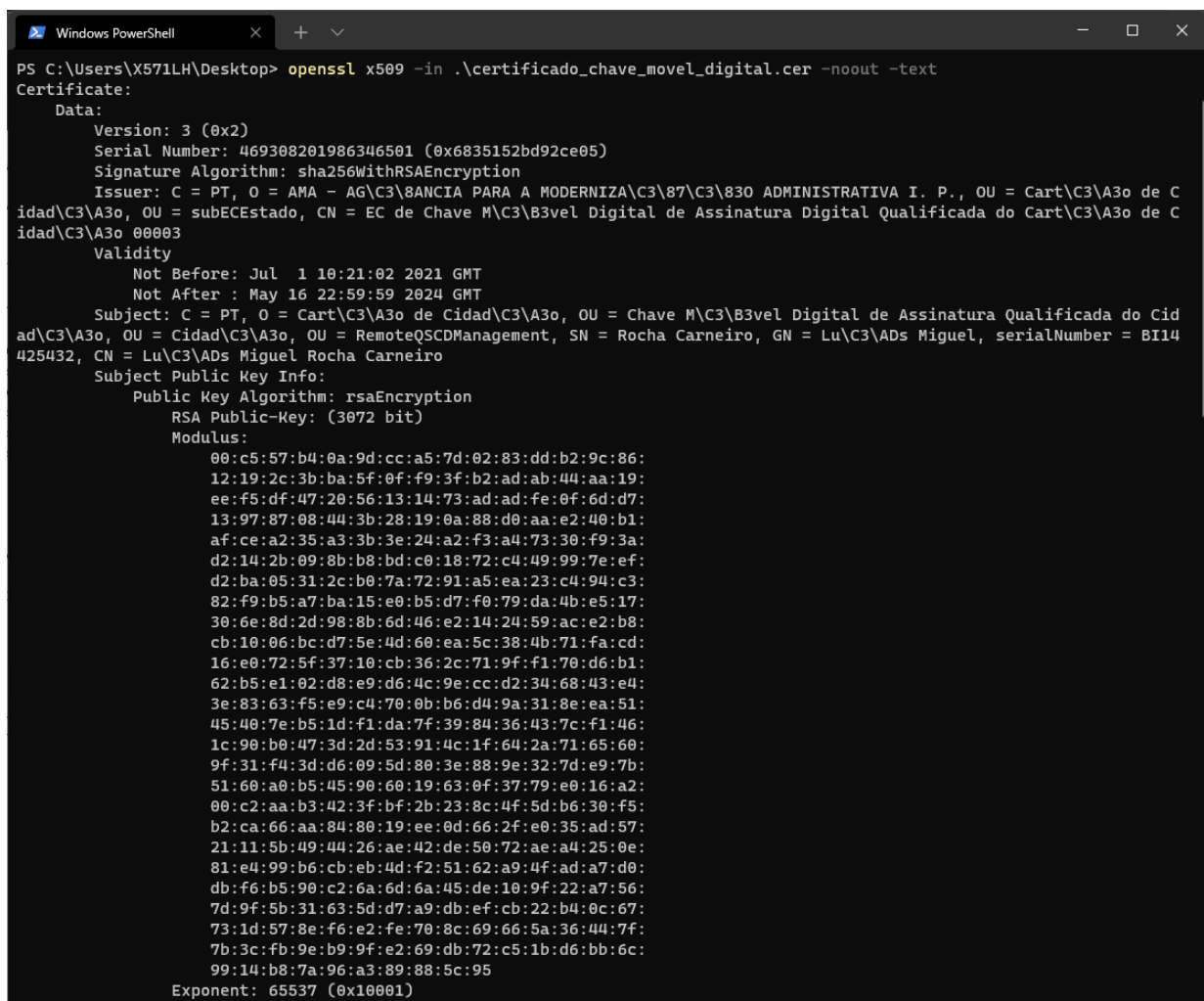
Capítulo 1

Parte VIII: Infraestrutura de chave pública

1.1 Pergunta P.VIII.1.1

1. Utilize o openssl para ver o conteúdo do seu certificado CC/CMD. Identifique as várias componentes do mesmo, e o seu significado e/ou objetivo.

O comando utilizado, assim como o respetivo output, pode ser visto nas imagens seguintes.



```
PS C:\Users\X571LH\Desktop> openssl x509 -in .\certificado_chave_movel_digital.cer -noout -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 469308201986346501 (0x6835152bd92ce05)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = PT, O = AMA - AG\C3\BANCIA PARA A MODERNIZA\C3\87\C3\830 ADMINISTRATIVA I. P., OU = Cart\C3\A3o de C
        idad\C3\A3o, OU = subECEstado, CN = EC de Chave M\C3\B3vel Digital de Assinatura Digital Qualificada do Cart\C3\A3o de C
        idad\C3\A3o 00003
        Validity
            Not Before: Jul  1 10:21:02 2021 GMT
            Not After : May 16 22:59:59 2024 GMT
        Subject: C = PT, O = Cart\C3\A3o de Cidad\C3\A3o, OU = Chave M\C3\B3vel Digital de Assinatura Qualificada do Cid
        ad\C3\A3o, OU = Cidad\C3\A3o, OU = RemoteQSCDManagement, SN = Rocha Carneiro, GN = Lu\C3\ADS Miguel, serialNumber = BI14
        425432, CN = Lu\C3\ADS Miguel Rocha Carneiro
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public-Key: (3072 bit)
            Modulus:
                00:c5:57:b4:0a:9d:cc:a5:7d:02:83:dd:b2:9c:86:
                12:19:2c:3b:ba:5f:0f:f9:3f:b2:ad:ab:44:aa:19:
                ee:f5:df:47:20:56:13:14:73:ad:ad:fe:0f:6d:d7:
                13:97:87:08:44:3b:28:19:0a:88:d0:aa:e2:40:b1:
                af:ce:a2:35:a3:3b:3e:24:a2:f3:a4:73:30:f9:3a:
                d2:14:2b:09:8b:b8:bd:c0:18:72:c4:49:99:7e:ef:
                d2:ba:05:31:2c:b0:7a:72:91:a5:ea:23:c4:94:c3:
                82:f9:b5:a7:ba:15:e0:b5:d7:f0:79:da:4b:e5:17:
                30:6e:8d:2d:98:8b:6d:46:e2:14:24:59:ac:e2:b8:
                cb:10:06:bc:d7:5e:4d:60:ea:5c:38:4b:71:fa:cd:
                16:e0:72:5f:37:10:cb:36:2c:71:9f:f1:70:d6:b1:
                62:b5:e1:02:d8:e9:d6:4c:9e:cc:d2:34:68:43:e4:
                3e:83:63:f5:e9:c4:70:0b:b6:d4:9a:31:8e:ea:51:
                45:40:7e:b5:1d:f1:da:7f:39:84:36:43:7c:f1:46:
                1c:90:b0:47:3d:2d:53:91:4c:1f:64:2a:71:65:60:
                9f:31:f4:3d:d6:09:5d:80:3e:88:9e:32:7d:e9:7b:
                51:60:a0:b5:45:90:60:19:63:0f:37:79:e0:16:a2:
                00:c2:aa:b3:42:3f:bf:2b:23:8c:4f:5d:b6:30:f5:
                b2:ca:66:aa:84:80:19:ee:0d:66:2f:e0:35:ad:57:
                21:11:5b:49:44:26:ae:42:de:50:72:ae:a4:25:0e:
                81:e4:99:b6:cb:eb:4d:f2:51:62:a9:4f:ad:a7:d0:
                db:f6:b5:90:c2:6a:6d:6a:45:de:10:9f:22:a7:56:
                7d:9f:5b:31:63:5d:d7:a9:db:ef:cb:22:b4:0c:67:
                73:1d:57:8e:f6:e2:fe:70:8c:69:66:5a:36:44:7f:
                7b:3c:fb:9e:b9:9f:e2:69:db:72:c5:1b:d6:bb:6c:
                99:14:b8:7a:96:a3:89:88:5c:95
            Exponent: 65537 (0x10001)
```

Figura 1.1: Print do conteúdo do certificado da Chave Móvel Digital (1)

```
Windows PowerShell
X509v3 extensions:
  X509v3 Basic Constraints: critical
    CA:FALSE
  X509v3 Authority Key Identifier:
    keyid:6A:C6:E5:B3:7A:BC:06:74:EF:E7:90:A6:96:D8:70:C3:B7:9A:83:A4

  Authority Information Access:
    OCSP - URI:http://ocsp.cmd.cartaodecidadao.pt/publico/ocsp

  X509v3 Freshest CRL:
    Full Name:
      URI:http://pki.cartaodecidadao.pt/publico/lrc/cc_sub-ec_cidadao_cmd_crl0003_delta_p0003.crl

  X509v3 Certificate Policies:
    Policy: 2.16.620.1.1.1.2.10
      CPS: http://www.scee.gov.pt/pcert
    Policy: 0.4.0.194112.1.2
      Policy: 2.16.620.1.1.1.2.4.3.0.1.1
        CPS: http://pki.cartaodecidadao.pt/publico/politicas/pc/cc_sub-ec_cidadao_cmd_pc.html
      Policy: 2.16.620.1.1.1.2.4.3.0.7
        CPS: http://pki.cartaodecidadao.pt/publico/politicas/dpc/cc_sub-ec_cidadao_cmd_dpc.html
      Policy: 0.4.0.2042.1.2

  X509v3 Subject Directory Attributes:
    0.0...+.....1...20000529120000Z
  qcStatements:
    0..0.....F..0.....F..0.....F...0.....F..0...07.9https://pki.cartaodecidadao.pt/publico/politicas/cps.html..PT07.9https://pki.cartaodecidadao.pt/publico/politicas/cps.html..EN
  X509v3 CRL Distribution Points:
    Full Name:
      URI:http://pki.cartaodecidadao.pt/publico/lrc/cc_sub-ec_cidadao_cmd_crl0003_p0003.crl

  X509v3 Subject Key Identifier:
    B5:BC:E9:C0:50:61:3F:F1:72:B1:DD:E1:09:60:86:73:27:D6:DC:9F
  X509v3 Key Usage: critical
    Non Repudiation
  Signature Algorithm: sha256WithRSAEncryption
    5b:a6:e1:7f:7a:75:5f:b0:79:0f:42:26:4d:c7:f5:b9:9e:3c:
    46:b3:91:2a:a9:db:e4:ab:80:e3:ce:c2:14:9d:03:d9:56:e3:
    ef:bf:9e:09:d5:ae:22:7a:57:e7:b6:a4:0a:be:97:c2:05:08:
    ca:c8:b5:ef:b9:a4:02:84:59:4f:bf:06:eb:16:7a:04:5b:91:
    59:85:eb:d5:da:27:11:ad:07:9b:b5:ee:ae:f6:4f:f6:96:16:
    6a:6f:4d:a7:2d:dd:92:ec:75:a3:b7:8e:9f:f6:05:3f:9b:03:
    be:62:69:5a:81:1c:0b:5e:03:4c:d4:eb:22:3a:75:ef:d2:c7:
```

Figura 1.2: Print do conteúdo do certificado da Chave Móvel Digital (2)

```
Windows PowerShell
X509v3 Subject Directory Attributes:
  0.0...+.....1...20000529120000Z
qcStatements:
  0..0.....F..0.....F..0.....F..0.....F..0.....F..0..07.9https://pki.cartaodecidadao.pt/publico/politicas/cps.html..PT07.9https://pki.cartaodecidadao.pt/publico/politicas/cps.html..EN
X509v3 CRL Distribution Points:

  Full Name:
    URI:http://pki.cartaodecidadao.pt/publico/lrc/cc_sub-ec_cidadao_cmd_crl0003_p0003.crl

X509v3 Subject Key Identifier:
  B5:BC:E9:C0:50:61:3F:F1:72:B1:DD:E1:09:60:86:73:27:D6:DC:9F
X509v3 Key Usage: critical
  Non Repudiation
Signature Algorithm: sha256WithRSAEncryption
5b:a6:e1:7f:7a:75:5f:b0:79:0f:42:26:4d:c7:f5:b9:9e:3c:
46:b3:91:2a:a9:db:e4:ab:80:e3:ce:c2:14:9d:03:d9:56:e3:
ef:bf:9e:09:d5:ae:22:7a:57:e7:b6:a4:0a:be:97:c2:05:08:
ca:c8:b5:ef:b9:a4:02:84:59:4f:bf:06:eb:16:7a:04:5b:91:
59:85:eb:d5:da:27:11:ad:07:9b:b5:ee:ae:f6:4f:f6:96:16:
6a:6f:4d:a7:2d:dd:92:ec:75:a3:b7:8e:9f:f6:05:3f:9b:03:
be:62:69:5a:81:1c:0b:5e:03:4c:d4:eb:22:3a:75:ef:d2:c7:
51:27:ec:19:e5:24:4b:97:4b:d5:0b:7c:55:66:2f:3c:0c:cf:
6b:b3:c3:f9:4a:36:66:7b:5e:53:5b:25:66:bb:c3:6b:25:08:
f1:e9:b1:a7:ec:20:92:fd:3e:fb:24:3e:3a:4e:7d:03:9c:49:
97:71:93:45:56:0c:cb:c0:e7:d0:e1:64:4c:7d:9c:fa:60:51:
5f:e9:8c:a6:15:22:64:d6:7d:ed:75:6d:0f:c4:e4:92:bb:b7:
5c:7a:bf:71:3f:87:2c:5f:fd:e8:12:69:8a:63:cd:9a:be:45:
5f:12:24:1f:e7:36:bf:56:28:a5:f9:75:fd:f5:d2:20:66:9b:
6c:c8:3b:d2:86:27:d3:26:51:0f:ce:27:62:33:73:b6:6f:ba:
d2:56:1d:cf:69:81:87:70:3a:6b:31:67:19:a1:74:7e:dc:e0:
f1:02:ff:58:6d:c9:3c:4b:5f:33:ef:2f:f6:d4:f9:5c:18:9e:
42:f6:8c:f7:7e:ba:2a:3a:65:36:3f:71:d6:23:f2:6b:4e:dd:
44:18:7b:e2:73:99:41:cb:39:3b:44:c9:9e:91:b6:9d:b4:09:
bf:8d:f8:80:8a:c7:92:1a:52:82:24:fa:d5:f6:d2:4e:b5:2a:
c7:40:8d:c6:39:a0:18:e1:31:38:d1:9e:20:52:06:82:f6:d9:
5a:f2:a0:c0:46:ed:25:d5:ef:e1:50:9b:84:24:5a:d2:b9:20:
a7:aa:88:81:cd:fd:37:e5:06:85:65:89:89:88:28:56:2f:7e:
3f:d4:c2:43:22:a5:3d:85:93:d9:80:e0:38:1b:1d:6c:c9:de:
c9:22:c3:f8:1c:81:3c:f8:2c:48:92:22:39:9d:71:af:40:79:
ef:69:64:b5:68:2a:e0:cf:9e:93:ea:9b:58:c5:a7:9a:19:e0:
87:00:fc:1e:b7:b9:f9:cb:a7:7c:22:00:b3:b8:93:42:7f:9a:
61:28:3d:f4:6a:a9:02:c5:5b:ce:2e:79:28:89:5f:f9:8b:0d:
7b:49:d3:2e:b3:db:ae:1b
PS C:\Users\X571LH\Desktop>
```

Figura 1.3: Print do conteúdo do certificado da Chave Móvel Digital (3)

É possível observar que o certificado está dividido em duas secções: os dados e o algoritmo de assinatura. Excetuando a subsecção das extensões do X509v3 (devido à sua extensão), os dados contêm:

- a versão do certificado - 3.
- o *Serial Number* - 469308201986346501 (0x6835152bd92ce05).
- o algoritmo de assinatura utilizado, nomeadamente o SHA-256 com RSA (sha256WithRSAEncryption).
- o país, organização, unidades organizacionais (neste caso, o certificado apresenta duas) e o "*common name*" (nome associado) ao emissor. Ou seja, o "EC de Chave Móvel Digital de Assinatura Digital Qualificada do Cartão de Cidadão 00003" da organização "AMA - AGÊNCIA PARA A MODERNIZAÇÃO ADMINISTRATIVA I. P." de Portugal, com as unidades organizacionais "subECEstado" e "Cartão de Cidadão".
- a validade do certificado - é válido desde 1 de julho de 2021 das 10:21:02 GMT até 16 de maio de 2024 até às 23:59:59 GMT.
- o sujeito, neste caso um dos membros do grupo. Ao final de contas, o que se pretende validar é a sua assinatura usando a chave móvel digital. Concretamente, a assinatura de Luís Miguel Rocha Carneiro, cidadão português com o BI N^o 14425432.
- a chave pública do sujeito mencionado, que foi cifrada com RSA e tem 3072 bits, cujo módulo e expoente se podem observar na figura 1.1.

Acerca das extensões do X509v3, apenas a primeira e última são críticas, ou seja, caso uma aplicação encontre estas extensões e não as reconheça, deve rejeitar o certificado. Explicando as extensões por ordem é possível encontrar:

- que o sujeito do certificado não é uma autoridade de certificação (claramente não, pois o sujeito é um membro do grupo).
- o identificador da chave de autoridade.
- o URI do OCSP, ou seja, onde obter a informação relativamente à revogação do certificado.
- o URI do CRL mais recente, com o mesmo objetivo do ponto anterior.
- as políticas do certificado, que podem ter o CPS associado (declaração de práticas de certificação). Estas definem a aplicabilidade de certificados a uma determinada comunidade ou classe de aplicações com requisitos de segurança comuns.
- os atributos do *Subject Directory*, ou seja, atributos adicionais associados com o sujeito, como complemento do campo do sujeito e da extensão do nome alternativo.
- o identificador da chave do sujeito.
- a finalidade do uso da chave presente no certificado; neste caso, o não repúdio.

Finalmente, o algoritmo de assinatura identifica (mais uma vez) que foi usado o SHA-256 com RSA e a assinatura em si.

2. Utilizando a(s) biblioteca(s) que achar mais adequada, desenvolva um program linha de comando em Python que tem como input um certificado (neste caso, o exemplo que terá que testar é com o seu certificado CC/CMD, mas deve funcionar para o caso geral de certificados CC/CMD), e vai indicar se o mesmo está ou não revogado através da consulta da CRL.

As bibliotecas utilizadas neste programa foram *requests* para transferir o ficheiro CRL, *OpenSSL* para ler e verificar o certificado como o ficheiro CRL e *sys* para receber o certificado como parâmetro quando o programa é executado.

O programa recebe como argumento de entrada o caminho da chave móvel digital e chama a função *VerificarCertificado*. Nesta função o programa lê a chave móvel digital e verifica se já expirou a data de validade.

```
48 def VerificarCertificado(certificado):
49     cert= certificado.read()
50
51     hasExpired = crypto.load_certificate(crypto.FILETYPE_PEM, cert).has_expired()
52     ExpireDate = str(crypto.load_certificate(crypto.FILETYPE_PEM, cert).get_notAfter())
53     full = crypto.load_certificate(crypto.FILETYPE_PEM, cert)
54
55     if(hasExpired):
56         print('Segundo a chavemovel.cer o certificado expirou')
57     else:
58         print('Segundo a chavemovel.cer o certificado não expirou')
59
60     print('O certificado expira no ano: ' + ExpireDate[2:6] + ' mes: ' + ExpireDate[7:8] +
61         ' dia: ' + ExpireDate[9:10])
62
63     cert_dict = ssl._ssl._test_decode_cert('certificado_chave_movel_digital.cer')
64     splittedCert = SplitString(str(cert_dict))
65     crlURL = GetCRLURL(splittedCert)
66
67     print('O URL do ficheiro crl extraído do certificado é: \n' + crlURL)
68
69     crl = GetCRLFile(crlURL)
70
71     VerificarCRL(crl)
72
73
74 chaveDigitalMovel = open(sys.argv[1], 'r')
75 VerificarCertificado(chaveDigitalMovel)
```

Figura 1.4: Verificar Certificado

De seguida utiliza as funções *SplitString* e *GetCRLURL* para obter o URL do ficheiro CRL fornecido pelo certificado.

```
def SplitString(text):
    split = text.split()
    return split

def GetCRLURL(splittedCert):
    for split in splittedCert:
        if("crl" in split):
            url=split
    url = url.replace('\', ' ')
    url = url.replace(',', ' ')
    url = url.replace('}', ' ')
    url = url.replace(')', ' ')
    url = url.replace('(', ' ')
    return url
```

Figura 1.5: SplitString e GetCRLURL

Por último usa as funções *GetCRLFile* e *VerificarCRL* para transferir o ficheiro CRL e verificar se o certificado ainda está válido.

```
22 def GetCRLFile(url):
23     resp = requests.get(url)
24     crl = OpenSSL.crypto.load_crl(OpenSSL.crypto.FILETYPE_ASN1, resp.content)
25     return crl
26
27
28 def VerificarCRL(crl):
29     # Exportar CRL como um cryptography CRL.
30     crl_crypto = crl.to_cryptography()
31     # Ler o certificado
32     with open(sys.argv[1], "r") as f:
33         cert_buf = f.read()
34
35     ca = OpenSSL.crypto.load_certificate(OpenSSL.crypto.FILETYPE_PEM, cert_buf)
36     # Obter chave publica
37     ca_pub_key = ca.get_pubkey().to_cryptography_key()
38
39     # Validar CRL sobre Certificado
40     valid_signature = crl_crypto.is_signature_valid(ca_pub_key)
41
42     if(valid_signature):
43         print('A assinatura é valida')
44     else:
45         print('A assinatura não é valida')
```

Figura 1.6: GetCRLFile e VerificarCRL

O output final do programa é o seguinte:

```
[obsession@ARES Desktop]$ python chavedigitaltest.py 'certificado_chave_movel_digital.cer'
Segundo a chavemovel.cer o certificado não expirou
O certificado expira no ano: 2024 mes: 5 dia: 6
O URL do ficheiro crl extraído do certificado é:
http://pki.cartaodecidadao.pt/publico/lrc/cc_sub-ec_cidadao_cmd_crl0003_p0003.crl
A assinatura não é valida
```

Figura 1.7: Output

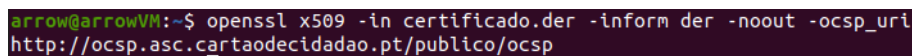
Infelizmente a assinatura é dada como inválida quando deveria ser válida.

Para verificar se o certificado ainda não foi revogado tem de se ver se o número de série do certificado se encontra na lista do ficheiro CRL. Caso este não esteja lá, então ainda é válido.

3. Utilizando a(s) biblioteca(s) que achar mais adequada, desenvolva um program linha de comando em Python que tem como input um certificado (neste caso, o exemplo que terá que testar é com o seu certificado CC/CMD, mas deve funcionar para o caso geral de certificados CC/CMD), e vai indicar se o mesmo está ou não revogado através da consulta do OCSP.

De modo a ser possível extrair o URL do serviço de OCSP, utilizou-se o seguinte comando do openssl:

```
openssl x509 -in cert.der -inform der -noout -ocsp_uri
```



```
arrow@arrowVM:~$ openssl x509 -in certificado.der -inform der -noout -ocsp_uri
http://ocsp.asc.cartaodecidadao.pt/publico/ocsp
```

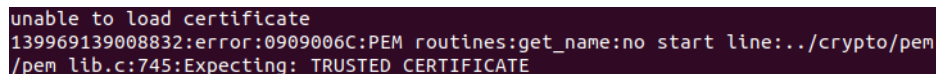
Figura 1.8: Extração do URL do serviço de OCSP do certificado

Para a construção dos *requests* ao serviço de OCSP, escolheu-se a biblioteca `cryptography`, nomeadamente o módulo `X.509`, que contém métodos de *load* de certificados e de construção de *requests* (`OCSPRequestBuilder`).

A partir deste *request* iria-se obter as informações pretendidas - a data de verificação, o URL do serviço e a validade do certificado.

Problemas de Implementação

Na resolução da alínea 3 a equipa de trabalho deparou-se com algumas dificuldades na construção do *request* ao serviço. Uma vez que era pretendido utilizar a classe `OCSPRequestBuilder` para construir o request, não foi possível encontrar uma solução na qual se conseguisse utilizar o URL extraído do certificado para obter a informação pretendida. Para além disto, a equipa de trabalho também se deparou com dificuldades em obter o *issuer* assim como o seu certificado, necessários para completar o *request*. Para além disto, a equipa também se deparou com o seguinte erro, que impossibilitou o *load* do certificado em si:



```
unable to load certificate
139969139008832:error:0909006C:PEM routines:get_name:no start line:../crypto/pem
/pem_lib.c:745:Expecting: TRUSTED CERTIFICATE
```

Figura 1.9: Erro no carregamento do certificado

Infelizmente, o grupo de trabalho foi incapaz de detetar a causa.

Capítulo 2

Referências

- Alínea 1:
 - <https://crypto.stackexchange.com/questions/35608/x509-certificate>
 - <https://datatracker.ietf.org/doc/html/rfc3739>
 - <https://datatracker.ietf.org/doc/html/rfc3280#page-28>
- Alínea 2:
 - <https://www.pyopenssl.org/en/stable/api.html>
- Alínea 3:
 - https://www.mksssoftware.com/docs/man1/openssl_x509.1.asp
 - https://docs.microfocus.com/NNMi/10.30/Content/Administer/NNMi_Deployment/Advanced_Configurations/Validate_Cert_Using_OCSP.htm
 - <https://cryptography.io/en/latest/x509/reference/>
 - <https://www.ibm.com/docs/en/datapower-gateway/7.6?topic=functions-ocsp-validate-certificate>