(a) Erstellen Sie eine Textdatei mit dem Inhalt

das ist ein test

das ist ein test

das ist ein test

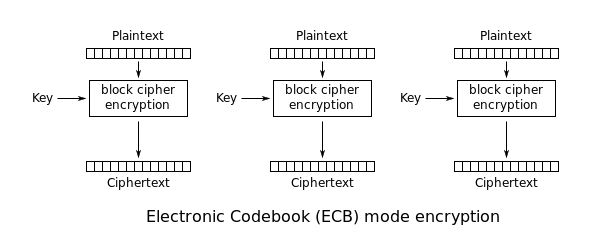
das ist ein test

bei dem sich der Satz etwa 50 mal wiederholt. Verschlüsseln Sie diese Datei mittels AES im ECB-Mode bei einer Schlüssellänge von 192 Bit und können Sie den Geheimtext. Was fällt Ihnen am Geheimtext auf und wie lässt sich das Ergebnis interpretieren? Wie lässt sich ein solcher Geheimtext vermeiden?

**openssl aes-192-ecb -in text.txt -out cipher.txt**

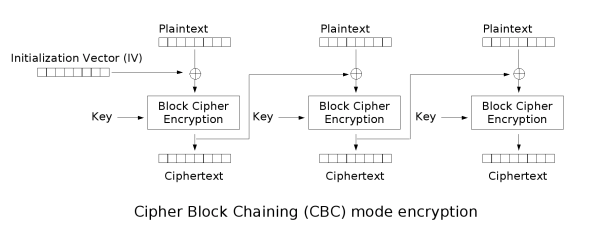
The ciphertext will repeat with a special block.

AES-ECB mode divides the plaintext to 128-bit Blocks, each block encrypted with a special key. When the plaintext repeat with a special block, then the ciphertext will repeat with a special block.



The image from wikipedia

we can avoid this problem if the special key can be changed when encryption. For example, AES-CBC mode.



The image from wikipedia

(b) Erzeugen Sie ein RSA-Schlüsselpaar mit Moduluslänge 2048 Bit und speichern Sie den Schlüssel im PEM-Format in der Datei rsaprivkey.pem ab. Das Schlüsselpaar soll bei der Erstellung mit einer 192 Bit AES Verschlüsselung geschützt werden.

**openssl genrsa -ase192 -out rsaprivkey.pem -passout pass:11111 2048**

(c) Geben Sie den RSA-Schlüssel im lesbaren Textformat auf dem Bildschirm aus und erläutern Sie die Ausgabe.

**Way1:**

**Way2(faster):**

faster than)

Reference: <http://www.di-mgt.com.au/crt_rsa.html>

**Ergebnis:**

Private-Key: (2048 bit)

modulus: N = p \* q

00:c7:46:3c:72:29:3b:54:4a:6d:54:f9:2c:d7:c0:

11:91:f2:0d:ae:6d:74:18:8a:28:82:26:6c:a1:2c:

bd:1d:61:ba:3e:56:2d:57:71:89:36:4d:d6:85:6c:

7d:d1:9b:51:23:e3:28:04:7f:4b:70:e9:11:1c:e7:

7a:cb:26:9a:df:1f:9e:73:03:44:3e:9e:5f:4a:85:

8b:19:42:73:9f:d7:a7:35:0d:3d:aa:7d:98:78:b2:

e3:a2:40:71:74:19:a9:77:68:10:34:5b:5d:7e:3e:

d1:8d:ba:10:45:61:67:7b:97:bc:db:c1:5b:61:ee:

4f:e0:e4:e3:ad:53:b1:48:1e:a8:46:57:94:59:be:

59:bf:78:f8:c8:89:a7:93:4b:72:92:f1:ad:10:f5:

9d:00:e7:f6:6f:91:75:e1:32:3c:0c:3c:d7:82:b8:

96:4e:fc:62:0d:37:48:0b:75:44:18:df:97:f3:bd:

5b:44:0d:3f:0b:6f:f8:0e:97:54:29:06:48:b6:8d:

d1:e3:95:a3:c2:15:54:e6:f5:7c:47:46:3d:1a:ab:

83:1b:a5:1a:83:52:ae:59:2a:de:45:d8:f2:23:57:

d9:0e:57:df:a8:7d:c0:44:01:15:a9:7c:dd:1c:52:

50:dc:af:c5:90:f7:fa:cb:d2:5d:d5:76:23:d3:60:

54:97

publicExponent: 65537 (0x10001) Public key (often 65537)

privateExponent: Private key d, ()

00:b2:8b:db:dc:ed:d4:9a:d1:94:e4:6d:de:1f:f1:

97:cc:8c:74:75:b4:dc:30:a0:99:6b:03:c2:91:26:

e4:cb:4e:a4:f0:f5:85:ea:b3:6f:9d:98:60:7d:6b:

97:d0:16:b9:bc:cd:eb:2d:1c:61:1b:e9:b7:cb:8e:

1c:17:a9:9f:5a:0b:9a:8c:b2:80:93:1a:93:86:d1:

e0:af:e1:3a:78:1d:61:53:c7:69:59:38:ef:44:36:

04:7f:7b:fe:2b:68:52:cf:0a:0e:7a:da:e2:0a:1b:

7e:e6:dd:af:76:7a:af:9d:ad:77:29:ef:35:51:a9:

95:28:8b:ec:f3:7f:e9:78:13:6a:8f:4c:4b:6c:a9:

dc:bf:bf:c8:63:b6:f5:47:58:bb:13:7c:02:3e:b0:

38:9c:78:5e:0b:98:e0:02:ed:af:25:ab:46:d4:ca:

cb:26:20:42:81:74:fd:bb:7c:aa:8c:bd:4b:cd:0d:

48:b9:da:ef:d8:6f:11:0b:77:07:5f:cf:ea:8d:a1:

56:91:b7:77:a8:7d:12:c0:63:43:e2:69:81:e3:98:

05:92:59:a8:a8:dd:34:56:9c:78:da:a1:64:3f:79:

e5:f2:12:a6:2e:ca:18:37:68:2b:aa:55:3b:e4:a2:

da:26:76:eb:54:e8:5b:0f:49:55:51:2a:2f:d9:55:

d7:01

prime1: p

00:ee:f4:70:cb:f0:56:ae:41:e6:19:d7:6a:38:d4:

5e:39:f0:54:77:b5:56:4d:4f:c1:42:c4:17:40:dc:

b4:19:4f:03:93:e3:a2:2c:e9:0e:8a:28:c8:b6:05:

a5:37:1e:4d:ff:5e:3e:1c:df:cd:93:c9:18:f2:ec:

c8:b1:42:79:1f:cb:c0:1e:2b:eb:39:4b:4d:a6:44:

b9:22:bf:9e:b8:b8:c6:e1:a8:98:be:22:cc:ac:f2:

9a:b6:4f:fb:b5:5b:49:5d:57:65:67:ba:d6:06:c2:

13:4a:a3:95:33:85:29:b1:0c:40:ba:09:3a:86:a2:

0d:ab:97:17:0a:37:b9:ec:ab

prime2: q

00:d5:7d:30:77:86:e3:5f:b1:e7:a8:3d:28:88:01:

21:f1:fd:79:42:7e:b2:bd:b5:ae:54:3a:5f:ed:bf:

18:c5:e1:85:3e:0a:6e:dc:6b:75:b0:1f:e6:69:a1:

f2:c3:23:21:bd:ed:6b:0f:34:cc:ab:bb:94:15:7a:

80:02:a4:e1:60:df:08:6a:cf:75:eb:5e:f9:16:db:

c1:d7:7b:75:6a:18:f4:c6:18:5e:d5:2e:69:f9:eb:

94:f8:b4:99:4d:19:ad:80:20:72:d9:db:96:65:19:

04:19:e7:a0:9f:fd:b4:23:9c:2e:cb:06:77:e8:8c:

f0:c2:de:87:d5:17:ed:9f:c5

exponent1: dP = d mod (p - 1)

77:11:36:63:68:8e:c8:a5:ed:92:52:d4:8b:1c:c7:

16:f7:ba:1f:2c:2c:a4:b9:69:03:6b:d2:6a:6f:8f:

0e:4d:60:eb:00:32:7f:4e:62:00:55:f2:2d:41:02:

e2:44:27:79:67:ec:6b:56:d9:89:1a:5b:64:30:b8:

90:1b:1f:5f:c8:97:22:71:9f:f4:1c:a1:c4:87:d9:

f8:86:a3:00:48:92:51:3f:41:5c:fc:da:20:13:83:

23:f5:16:e7:77:65:47:2d:95:49:ea:bf:5a:fe:b9:

be:27:64:e1:17:29:7f:1f:2b:e1:87:57:b2:5e:6d:

91:62:e8:08:1f:5c:75:a1

exponent2: dQ = d mod (q - 1)

4d:9b:84:d2:ee:a6:19:53:82:f3:15:c8:76:28:73:

3e:94:f6:4f:c5:df:14:37:c7:77:b7:00:c8:5c:7f:

94:ef:b6:90:27:1a:c5:e6:0a:92:4c:06:66:c0:08:

90:07:65:6c:b5:b2:05:85:1c:22:11:be:73:50:64:

d8:91:37:9d:0f:76:ef:dd:1a:f0:c5:45:a7:99:9b:

8d:c9:c3:95:1e:1d:5d:25:de:d3:80:5b:e6:0e:4d:

c9:5b:5d:9f:39:86:68:dd:62:e2:ef:80:6c:9f:fb:

6d:af:56:24:d3:05:79:2c:7d:96:30:73:c6:85:f5:

bf:e1:45:5f:a1:a3:30:41

coefficient: qInv = (inverse of q) mod p

00:cf:a3:0b:49:9e:28:04:6a:62:91:76:44:89:4a:

bd:71:f3:2e:b5:ac:81:d8:d4:28:8c:ca:fb:ba:ef:

01:af:eb:26:16:6b:a3:4e:da:95:ac:1a:a4:9e:68:

a5:b5:e0:d5:a4:cc:9e:e6:3b:00:2f:3a:e1:1d:13:

38:35:c6:75:75:bf:ed:52:13:36:db:c3:4b:b9:f3:

c0:ac:10:f3:db:5c:f9:09:35:e1:e1:1f:b4:48:cf:

a5:16:3d:46:69:29:af:25:f1:c9:aa:b6:7b:10:62:

87:e0:93:98:67:fc:76:82:5f:8c:b3:79:19:e5:ea:

b8:dd:c5:bd:b4:6d:e5:48:2b

-----BEGIN RSA PRIVATE KEY-----

MIIEpAIBAAKCAQEAx0Y8cik7VEptVPks18ARkfINrm10GIoogiZsoSy9HWG6PlYt

V3GJNk3WhWx90ZtRI+MoBH9LcOkRHOd6yyaa3x+ecwNEPp5fSoWLGUJzn9enNQ09

qn2YeLLjokBxdBmpd2gQNFtdfj7RjboQRWFne5e828FbYe5P4OTjrVOxSB6oRleU

Wb5Zv3j4yImnk0tykvGtEPWdAOf2b5F14TI8DDzXgriWTvxiDTdIC3VEGN+X871b

RA0/C2/4DpdUKQZIto3R45WjwhVU5vV8R0Y9GquDG6Uag1KuWSreRdjyI1fZDlff

qH3ARAEVqXzdHFJQ3K/FkPf6y9Jd1XYj02BUlwIDAQABAoIBAQCyi9vc7dSa0ZTk

bd4f8ZfMjHR1tNwwoJlrA8KRJuTLTqTw9YXqs2+dmGB9a5fQFrm8zestHGEb6bfL

jhwXqZ9aC5qMsoCTGpOG0eCv4Tp4HWFTx2lZOO9ENgR/e/4raFLPCg562uIKG37m

3a92eq+drXcp7zVRqZUoi+zzf+l4E2qPTEtsqdy/v8hjtvVHWLsTfAI+sDiceF4L

mOAC7a8lq0bUyssmIEKBdP27fKqMvUvNDUi52u/YbxELdwdfz+qNoVaRt3eofRLA

Y0PiaYHjmAWSWaio3TRWnHjaoWQ/eeXyEqYuyhg3aCuqVTvkotomdutU6FsPSVVR

Ki/ZVdcBAoGBAO70cMvwVq5B5hnXajjUXjnwVHe1Vk1PwULEF0DctBlPA5Pjoizp

DoooyLYFpTceTf9ePhzfzZPJGPLsyLFCeR/LwB4r6zlLTaZEuSK/nri4xuGomL4i

zKzymrZP+7VbSV1XZWe61gbCE0qjlTOFKbEMQLoJOoaiDauXFwo3ueyrAoGBANV9

MHeG41+x56g9KIgBIfH9eUJ+sr21rlQ6X+2/GMXhhT4KbtxrdbAf5mmh8sMjIb3t

aw80zKu7lBV6gAKk4WDfCGrPdete+Rbbwdd7dWoY9MYYXtUuafnrlPi0mU0ZrYAg

ctnblmUZBBnnoJ/9tCOcLssGd+iM8MLeh9UX7Z/FAoGAdxE2Y2iOyKXtklLUixzH

Fve6HywspLlpA2vSam+PDk1g6wAyf05iAFXyLUEC4kQneWfsa1bZiRpbZDC4kBsf

X8iXInGf9ByhxIfZ+IajAEiSUT9BXPzaIBODI/UW53dlRy2VSeq/Wv65vidk4Rcp

fx8r4YdXsl5tkWLoCB9cdaECgYBNm4TS7qYZU4LzFch2KHM+lPZPxd8UN8d3twDI

XH+U77aQJxrF5gqSTAZmwAiQB2VstbIFhRwiEb5zUGTYkTedD3bv3RrwxUWnmZuN

ycOVHh1dJd7TgFvmDk3JW12fOYZo3WLi74Bsn/ttr1Yk0wV5LH2WMHPGhfW/4UVf

oaMwQQKBgQDPowtJnigEamKRdkSJSr1x8y61rIHY1CiMyvu67wGv6yYWa6NO2pWs

GqSeaKW14NWkzJ7mOwAvOuEdEzg1xnV1v+1SEzbbw0u588CsEPPbXPkJNeHhH7RI

z6UWPUZpKa8l8cmqtnsQYofgk5hn/HaCX4yzeRnl6rjdxb20beVIKw==

-----END RSA PRIVATE KEY-----

(d) Extrahieren Sie den öffentlichen RSA-Schlüssel aus rsaprivkey.pem und speichern Sie den Schlüssel im PEM-Format in der Datei rsapubkey.pem ab.

**openssl rsa -in rsaprivkey.pem -pubout -out rsapubkey.pem**

(e) Erzeugen Sie sich eine Textdatei plain.txt mit Ihrem Namen, Matrikelnummer und Studienfach und verschlüsseln Sie diese mittels RSA zur Datei cipher.bin.

**openssl rsautl -encrypt -inkey rsapubkey.pem -pubin -in plain.txt -out cipher.bin**

(f) Entschlüsseln Sie die Datei cipher.bin und vergleichen Sie den Klartext mit der ursprünglichen Textdatei.

**openssl rsautl -decrypt -inkey rsaprivkey.pem -in cipher.bin -out plain1.txt**

(g) Erzeugen Sie sich ein weiteres RSA-Schlüsselpaar, mit dem Sie Ihre Textdatei signieren und verifizieren. Warum ist es sinnvoll, für Verschlüsselung und Signatur zwei unterschiedliche RSA-Schlüsselpaare zu verwenden?

**openssl dgst -sign rsaprivkey3.pem -sha256 -out sign.txt plain.txt**

**openssl dgst -verify rsapubkey3.pem -sha256 -signature sign.txt plain.txt**

In my opinion, Signal private key updates to CA. But for Verschlüsselung the private key store in own PC.

(h) Lassen Sie sich das X.509-Zertifikat der Seite https://www.hs-emden-leer.de mittels eines geeigneten openssl-Befehls anzeigen und speichern Sie dann das Zertifikat im PEM-Format in der Datei hsel\_cert.pem ab.

**Openssl s\_client -showcerts -connect [www.hs-emden-leer.de:443](http://www.hs-emden-leer.de:443) | openssl x509 -out hsel\_cert.pem**

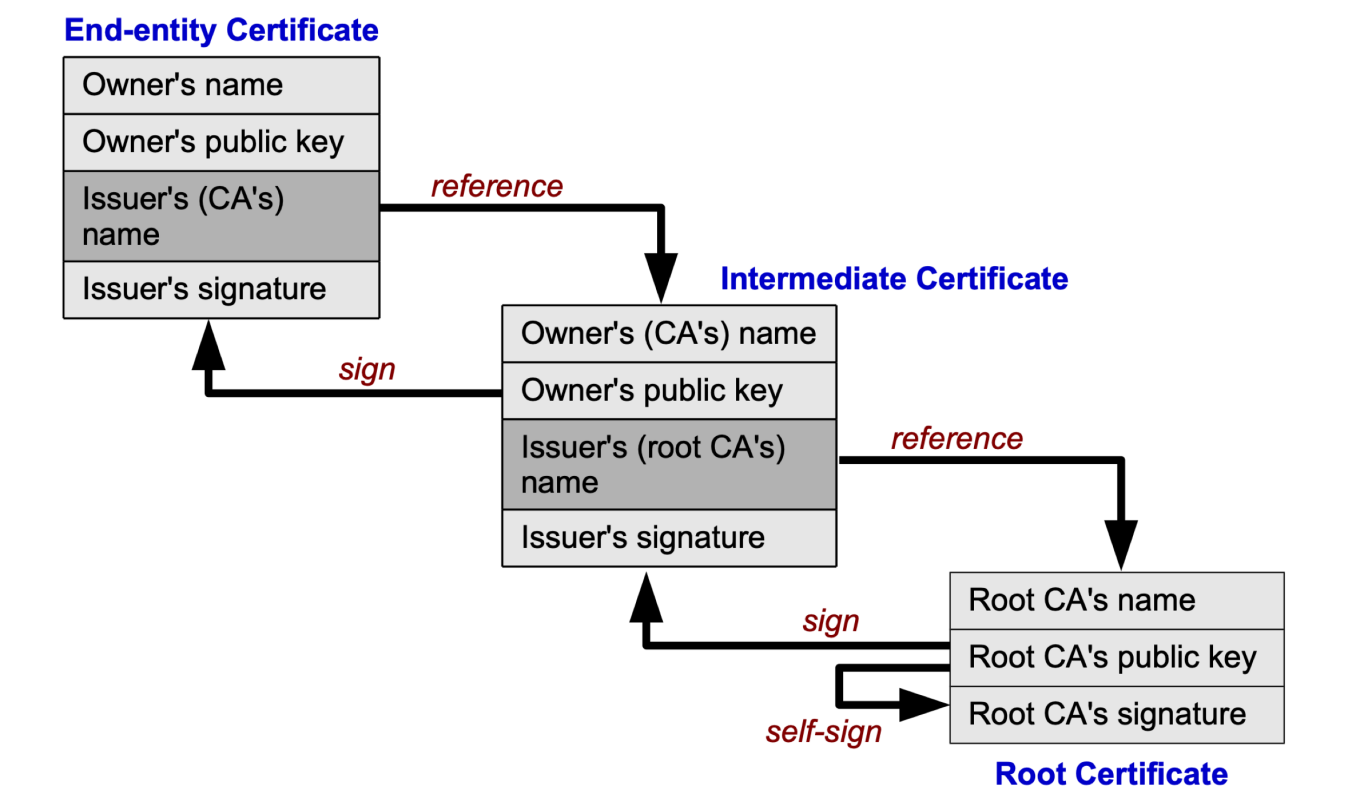
(i)Geben Sie die Zertifikatskette an. Was ist die Root Certification Authority (Root-CA)?

Deutsche Telekom Root CA 2

CA can sign the other companys or organization.

But who can sign the CA? other CA expect Root C?

Only Root CA can sign itself.



The image from wikipedia

(j) Wandeln Sie das Zertifikat der Datei hsel\_cert.pem in das DER-Format um und speichern Sie es in der Datei hsel\_cert.der. Geben Sie jeweils für die Datei hsel\_cert.pem sowie hsel\_cert.der die SHA1- und MD5-Hashwerte an. Sind diese im Zertifikat gespeichert? Sind die jeweiligen Werte für die beiden Kodierungen PEM und DER gleich?

**openssl x509 -in hsel\_cert.pem -out hsel\_cert.der**

die SHA1- und MD5- Hashwerte sind im Zertifikat nicht gespeichert. Die jewiligen Werte für die beiden Kodierungen PEM und DER sind gleich.

**openssl x509 -noout -fingerprint -sha1 -in hsel\_cert.pem(der)**

SHA1 Fingerprint=83:C1:61:DC:ED:15:A9:02:8D:69:04:25:FE:6A:58:24:C8:FB:01:A0

**openssl x509 -noout -fingerprint -md5 -in hsel\_cert.pem(der)**

MD5 Fingerprint=77:3F:DC:DF:9D:83:30:95:B9:F4:D0:B3:B2:64:E7:56

(k) Geben Sie das Zertifikat, abgespeichert in hsel\_cert.pem, im lesbaren Textformat auf dem Bildschirm aus.

**openssl x509 -in hsel\_cert.pem -text**

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

1a:bc:41:6c:f1:35:30

Signature Algorithm: sha256WithRSAEncryption

Issuer: C = DE, ST = Niedersachsen, O = Hochschule Emden/Leer, CN = HS-EL CA, emailAddress = pki@hs-emden-leer.de

Validity

Not Before: Jan 18 10:52:12 2016 GMT

Not After : Apr 16 10:52:12 2019 GMT

Subject: C = DE, ST = Niedersachsen, L = Emden, O = Hochschule Emden/Leer, OU = HRZ, CN = www.hs-emden-leer.de

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (2048 bit)

Modulus:

00:c1:27:9c:02:79:96:66:ee:ae:25:99:62:5d:0d:

6b:48:24:0f:52:96:5a:47:72:0e:3b:9e:3b:e0:f7:

f0:db:64:2b:6c:a2:a5:24:90:88:62:0b:d2:be:9a:

35:79:5e:24:e1:cd:27:9b:6b:22:05:1c:3f:96:37:

4b:49:5c:2e:7b:38:78:29:0d:d2:6c:02:ea:9b:75:

31:24:e0:d2:e5:8b:1b:a1:4c:20:26:e8:ad:0e:3f:

c1:63:c6:76:2c:76:9a:20:1a:52:03:99:e4:d9:10:

e5:17:92:76:fe:fb:7f:29:5b:ad:89:3e:e5:da:a7:

2f:59:0d:ec:1a:1a:15:7b:2e:3d:bf:97:ba:c6:71:

30:08:94:5c:3e:35:d3:2e:76:6c:83:63:7e:32:70:

de:d4:9d:e9:de:46:a6:95:b6:98:e9:0f:a4:de:6e:

92:dd:b8:2b:db:ec:90:8e:91:10:2c:19:bf:5d:bb:

ad:c5:70:41:83:71:c2:34:36:12:bc:75:74:81:aa:

17:0a:3c:08:b7:0d:84:27:0b:30:23:6e:e0:3b:77:

c0:7e:33:6a:39:45:c6:ae:d8:c1:51:67:cf:72:8e:

b7:5e:66:69:f8:85:e6:24:21:62:0c:7c:f0:35:07:

e4:1b:8f:4d:65:d0:de:bf:e6:81:9d:f7:62:55:60:

12:23

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Certificate Policies:

Policy: 1.3.6.1.4.1.22177.300.1.1.4.3.4

Policy: 1.3.6.1.4.1.22177.300.2.1.4.3.1

Policy: 1.3.6.1.4.1.22177.300.1.1.4

Policy: 1.3.6.1.4.1.22177.300.30

Policy: 2.23.140.1.2.2

X509v3 Basic Constraints:

CA:FALSE

X509v3 Key Usage:

Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment

X509v3 Extended Key Usage:

TLS Web Server Authentication

X509v3 Subject Key Identifier:

D0:BF:6A:70:AD:21:4E:B5:AB:64:9B:08:A9:AE:29:CE:69:F9:34:30

X509v3 Authority Key Identifier:

keyid:B5:8E:C1:94:C8:8B:7A:59:6E:41:8A:97:D4:1B:27:59:0B:D2:F2:CF

X509v3 Subject Alternative Name:

DNS:www.hs-emden-leer.de

X509v3 CRL Distribution Points:

Full Name:

URI:http://cdp1.pca.dfn.de/hs-emden-leer-ca/pub/crl/cacrl.crl

Full Name:

URI:http://cdp2.pca.dfn.de/hs-emden-leer-ca/pub/crl/cacrl.crl

Authority Information Access:

OCSP - URI:http://ocsp.pca.dfn.de/OCSP-Server/OCSP

CA Issuers - URI:http://cdp1.pca.dfn.de/hs-emden-leer-ca/pub/cacert/cacert.crt

CA Issuers - URI:http://cdp2.pca.dfn.de/hs-emden-leer-ca/pub/cacert/cacert.crt

Signature Algorithm: sha256WithRSAEncryption

18:ee:a2:e7:b8:ba:0c:28:e4:3e:95:f5:02:83:41:5e:3e:20:

1e:1b:7e:b7:d9:6b:b3:b9:a1:8c:3e:df:96:fa:2d:72:52:0a:

f6:b0:1a:9d:bb:64:58:66:b0:23:e1:aa:77:58:a9:ed:df:1a:

c8:b2:cb:3f:46:92:9a:69:83:95:e1:ee:11:98:da:ae:ac:f3:

62:37:c5:d3:0b:eb:85:3c:00:03:e6:e7:be:a1:b9:79:86:67:

c1:6c:4d:90:89:b0:d0:7c:f3:18:90:16:1b:41:c7:f9:32:21:

ed:32:e9:52:b7:60:d0:e8:1d:c2:ff:33:71:f1:f5:d3:47:46:

b2:08:e4:9b:68:6f:4e:47:fa:a3:dd:1e:de:48:02:62:e8:07:

a3:5f:96:a1:58:9c:d4:36:75:7c:5b:29:34:ea:84:a0:34:93:

14:7c:68:24:82:4a:4c:06:78:7b:14:01:b3:71:4a:a2:3a:37:

75:aa:0f:e6:0c:99:37:01:9d:a4:88:ae:ce:c5:28:49:de:a8:

93:92:6f:f3:51:ab:d2:62:2d:82:0d:45:fd:72:a5:a3:30:a7:

77:85:9e:b7:f6:a0:90:c1:d6:99:22:1f:2d:1a:8f:af:91:0c:

39:82:21:94:83:a0:8b:79:49:dc:2c:e1:14:08:87:de:48:03:

c3:63:66:ea

-----BEGIN CERTIFICATE-----

MIIF3jCCBMagAwIBAgIHGrxBbPE1MDANBgkqhkiG9w0BAQsFADB9MQswCQYDVQQG

EwJERTEWMBQGA1UECBMNTmllZGVyc2FjaHNlbjEeMBwGA1UEChMVSG9jaHNjaHVs

ZSBFbWRlbi9MZWVyMREwDwYDVQQDEwhIUy1FTCBDQTEjMCEGCSqGSIb3DQEJARYU

cGtpQGhzLWVtZGVuLWxlZXIuZGUwHhcNMTYwMTE4MTA1MjEyWhcNMTkwNDE2MTA1

MjEyWjCBgjELMAkGA1UEBhMCREUxFjAUBgNVBAgMDU5pZWRlcnNhY2hzZW4xDjAM

BgNVBAcMBUVtZGVuMR4wHAYDVQQKDBVIb2Noc2NodWxlIEVtZGVuL0xlZXIxDDAK

BgNVBAsMA0hSWjEdMBsGA1UEAwwUd3d3LmhzLWVtZGVuLWxlZXIuZGUwggEiMA0G

CSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDBJ5wCeZZm7q4lmWJdDWtIJA9SllpH

cg47njvg9/DbZCtsoqUkkIhiC9K+mjV5XiThzSebayIFHD+WN0tJXC57OHgpDdJs

AuqbdTEk4NLlixuhTCAm6K0OP8FjxnYsdpogGlIDmeTZEOUXknb++38pW62JPuXa

py9ZDewaGhV7Lj2/l7rGcTAIlFw+NdMudmyDY34ycN7UneneRqaVtpjpD6TebpLd

uCvb7JCOkRAsGb9du63FcEGDccI0NhK8dXSBqhcKPAi3DYQnCzAjbuA7d8B+M2o5

Rcau2MFRZ89yjrdeZmn4heYkIWIMfPA1B+Qbj01l0N6/5oGd92JVYBIjAgMBAAGj

ggJbMIICVzBZBgNVHSAEUjBQMBEGDysGAQQBga0hgiwBAQQDBDARBg8rBgEEAYGt

IYIsAgEEAwEwDwYNKwYBBAGBrSGCLAEBBDANBgsrBgEEAYGtIYIsHjAIBgZngQwB

AgIwCQYDVR0TBAIwADALBgNVHQ8EBAMCBPAwEwYDVR0lBAwwCgYIKwYBBQUHAwEw

HQYDVR0OBBYEFNC/anCtIU61q2SbCKmuKc5p+TQwMB8GA1UdIwQYMBaAFLWOwZTI

i3pZbkGKl9QbJ1kL0vLPMB8GA1UdEQQYMBaCFHd3dy5ocy1lbWRlbi1sZWVyLmRl

MIGNBgNVHR8EgYUwgYIwP6A9oDuGOWh0dHA6Ly9jZHAxLnBjYS5kZm4uZGUvaHMt

ZW1kZW4tbGVlci1jYS9wdWIvY3JsL2NhY3JsLmNybDA/oD2gO4Y5aHR0cDovL2Nk

cDIucGNhLmRmbi5kZS9ocy1lbWRlbi1sZWVyLWNhL3B1Yi9jcmwvY2FjcmwuY3Js

MIHbBggrBgEFBQcBAQSBzjCByzAzBggrBgEFBQcwAYYnaHR0cDovL29jc3AucGNh

LmRmbi5kZS9PQ1NQLVNlcnZlci9PQ1NQMEkGCCsGAQUFBzAChj1odHRwOi8vY2Rw

MS5wY2EuZGZuLmRlL2hzLWVtZGVuLWxlZXItY2EvcHViL2NhY2VydC9jYWNlcnQu

Y3J0MEkGCCsGAQUFBzAChj1odHRwOi8vY2RwMi5wY2EuZGZuLmRlL2hzLWVtZGVu

LWxlZXItY2EvcHViL2NhY2VydC9jYWNlcnQuY3J0MA0GCSqGSIb3DQEBCwUAA4IB

AQAY7qLnuLoMKOQ+lfUCg0FePiAeG3632WuzuaGMPt+W+i1yUgr2sBqdu2RYZrAj

4ap3WKnt3xrIsss/RpKaaYOV4e4RmNqurPNiN8XTC+uFPAAD5ue+obl5hmfBbE2Q

ibDQfPMYkBYbQcf5MiHtMulSt2DQ6B3C/zNx8fXTR0ayCOSbaG9OR/qj3R7eSAJi

6AejX5ahWJzUNnV8Wyk06oSgNJMUfGgkgkpMBnh7FAGzcUqiOjd1qg/mDJk3AZ2k

iK7OxShJ3qiTkm/zUavSYi2CDUX9cqWjMKd3hZ639qCQwdaZIh8tGo+vkQw5giGU

g6CLeUncLOEUCIfeSAPDY2bq

-----END CERTIFICATE-----