Prof. Dr Georgios Raptis

RealCA\_HowTo Version 0.7, 14.09.2023 English

Create an almost "real" CA

* Revocation of a certificate is possible
* Issuing CA certificates, CRLSigner, OCSPSigner, user certificates for TLS authentication, TLS certificates for a web server
  + Auth certificates are compatible with the certificates of the Physicians ID cards in Germany, with the exception of the multivaluedRDN in the SubjectDN.
* Generate CRL
* Start OCSP server
* Query OCSP responder

A script is provided below for initialising the CA. The following instructions are for documentation purposes.

Create all directories and files defined in the ca.cfg. Adjust rights to protect the CA (rights management is not shown here).

Please adjust paths if necessary.

Please adjust the URL of the OCSP responder in the ocsp extension!

mkdir -p private

mkdir newcerts

mkdir certs

touch index.txt

echo "01" > serial

echo "01" > crlnumber

mkdir crl

mkdir csr

Generate root key and root certificate

openssl genrsa -aes256 -out private/cakey.pem 4096

openssl req -config ca.cfg -key private/cakey.pem -new -x509 -days 3650 -sha256 -extensions v3\_ca -out cacert.pem

Generate OCSP responder key and certificate

openssl genrsa -out private/ocsp.key.pem 3072

openssl req -config ca.cfg -key private/ocsp.key.pem -new -sha256 -out csr/ocsp.csr.pem

openssl ca -config ca.cfg -extensions ocsp -days 700 -notext -md sha256 -in csr/ocsp.csr.pem -out certs/ocsp.cert.pem

Create CRLSigner

openssl genrsa -out private/crlsigner.key.pem 3072

export EMAIL=test@example.com

openssl req -config ca.cfg -key private/crlsigner.key.pem -new -sha256 -out csr/crlsigner.csr.pem

openssl ca -config ca.cfg -extensions crlsigner -days 700 -notext -md sha256 -in csr/crlsigner.csr.pem -out certs/crlsigner.cert.pem

Initialise CRL, generate

openssl ca -config ca.cfg -gencrl -out crl/ca.crl.pem

Attention, direct CRL, signed by CA certificate!

Check CRL

openssl crl -in crl/ca.crl.pem -text

Generate Server (TLS) Certificate

openssl genrsa -out private/tlssigner.key.pem 3072

export EMAIL=test@example.com

export DOMAIN=example.com

openssl req -config ca.cfg -key private/tlssigner.key.pem -new -sha256 -out csr/tlssigner.csr.pem

openssl ca -config ca.cfg -extensions server -days 700 -notext -md sha256 -in csr/tlssigner.csr.pem -out certs/tlssigner.cert.pem

Please test with Apache or nginx!

Generate user certificate

export EMAIL=test@example.com

openssl genrsa -out private/user1.key.pem 3072

openssl req -config ca.cfg -key private/user1.key.pem -new -sha256 -out csr/user1.csr.pem

openssl ca -config ca.cfg -extensions userauth -days 700 -notext -md sha256 -in csr/user1.csr.pem -out certs/user1.cert.pem

Revocation of a certificate

openssl ca -config ca.cfg -revoke certs/user2.cert.pem

openssl crl -in crl/ca.crl.pem -text

Launch OCSP Responder

openssl ocsp -host im-vm-025.oth-regensburg.de -port 8080 -text -index /home/george/ID-Provider/index.txt -CA /home/george/ID-Provider/cacert.pem -rkey /home/george/ID-Provider/private/ocsp.key.pem -rsigner /home/george/ID-Provider/certs/ocsp.cert.pem

openssl ocsp -host <hostname> -port 8080 -text -index <path>/index.txt -CA <path>/cacert.pem -rkey <path>/ocsp.key.pem -rsigner <path>/ocsp.cert.pem

Query OCSP

openssl ocsp -CAfile cacert.pem -host im-vm-025.oth-regensburg.de -port 8080 -text -issuer cacert.pem -serial 4

Contents of the configuration file ca.cfg

Please adapt ocsp extension (URL).

[ca]

default\_ca = ID provider

[ID Provider]

dir = .

certs = $dir/certs

crl\_dir = $dir/crl

new\_certs\_dir = $dir/newcerts

database = $dir/index.txt

serial = $dir/serial

RANDFILE = $dir/private/.rand

crlnumber = $dir/crlnumber

crl = $dir/crl/ca.crl.pem

crl\_extensions = crl\_ext

default\_crl\_days = 30

name\_opt = ca\_default

cert\_opt = ca\_default

certificate = $dir/certs/cacert.pem

serial = $dir/serial

private\_key = $dir/private/cakey.pem

default\_md = sha256

default\_days = 365

policy = policy\_any

[ policy\_any ]

countryName = optional

stateOrProvinceName = optional

organizationName = optional

organizationalUnitName = optional

serialNumber = optional

SN = optional

GN = optional

commonName = optional

emailAddress = optional

[ req ]

default\_bits = 3072

distinguished\_name = req\_DN

string\_mask = utf8only

utf8 = yes

[ req\_DN ]

countryName = "Country Name (2 letter code)".

countryName\_default = DE

countryName\_min = 2

countryName\_max = 2

0.organizationName = "Organization Name (eg, company)".

0.organisationName\_default = eHealth OTH-Regensburg

#organizationalUnitName = "Organizational Unit Name(eg, section)"

#organizationalUnitName\_default = Test certificate

serialNumber = "Serial number in the DN".

SN = "Surname

SN\_max = 64

GN = "first name

GN\_max = 64

commonName = "Common Name (eg, CA name)"

commonName\_max = 64

commonName\_default = Test CA for eDocs

[ v3\_ca ]

basicConstraints = critical,CA:true,pathlen:0

keyUsage = critical, keyCertSign,cRLSign

authorityKeyIdentifier=keyid

subjectKeyIdentifier=hash

authorityInfoAccess = OCSP;URI:http://im-vm-025.oth-regensburg.de:8080

[ ocsp ]

basicConstraints = critical,CA:false

keyUsage = critical, nonRepudiation

authorityKeyIdentifier=keyid

subjectKeyIdentifier=hash

extendedKeyUsage = OCSPSigning

authorityInfoAccess = OCSP;URI:http://im-vm-025.oth-regensburg.de:8080

[ server ]

basicConstraints = critical,CA:false

keyUsage = critical, digitalSignature,keyEncipherment

authorityKeyIdentifier=keyid

subjectKeyIdentifier=hash

extendedKeyUsage = serverAuth

subjectAltName = @alt\_names

[ userauth ]

authorityKeyIdentifier=keyid

subjectKeyIdentifier=hash

keyUsage = critical, digitalSignature,keyEncipherment

certificatePolicies = @baekpol,1.2.276.0.76.4.75

subjectAltName = email:${ENV::EMAIL}

basicConstraints = critical,CA:false

1.3.36.8.3.3 = ASN1:SEQ:admission

extendedKeyUsage = clientAuth,emailProtection

authorityInfoAccess = OCSP;URI:http://im-vm-025.oth-regensburg.de:8080

[baekpol]

policyIdentifier = 1.2.276.0.76.4.145

CPS.1=http://www.e-arztausweis.de/policies/EE\_policy.html

[admission]

admauth = EXPLICIT:4,SEQ:authname

s0 = SEQ:admissions

[authname]

rdn1 = SET:authcseq

rdn2 = SET:authorgseq

[authcseq]

sq1 = SEQ:authc

[authc]

oid = OID:2.5.4.6

value = PRINTABLESTRING:EN

[authorgseq]

sq2 = SEQ:authorg

[authorg]

oid = OID:2.5.4.10

value = UTF8String:${ENV::ADMAUTH}

[admissions]

s1 = SEQ:profinfo

[profinfo]

s2 = SEQ:profitem

[profitem]

s3 = SEQ:profitem1

[profitem1]

s4 = SEQ:item

s5 = SEQ:profoid

s6 = PRINTABLESTRING:${ENV::TID}

[item]

s7 = UTF8String:${ENV::ADM}

[profoid]

s8 = OID:1.2.276.0.76.4.30

[ testadmission ]

1.3.36.8.3.3 = DER:30:18:30:16:30:14:30:12:30:10:30:0e:0c:0c:41:72:7a:74:2f:c3:84:72:7a:74:69:6e

[ crldp1 ]

fullname=URI:ldap://im-vm-025.oth-regensburg.de:389/CN=CRL,O=eHealth%20OTH-Regensburg,C=DE?certificateRevocationList?

base?objectClass=cRLDistributionPoint

CRLissuer=dirName:crlissuer

[ crlissuer ]

C=DE

O=eHealth Test OTH-Regensburg

CN=Test CRLSigner 1:PN

[ crlsigner ]

basicConstraints = critical,CA:false

keyUsage = critical, cRLSign

authorityKeyIdentifier=keyid

subjectKeyIdentifier=hash

[ crl\_ext ]

authorityKeyIdentifier=keyid:always

[ alt\_names ]

DNS.1 = ${ENV::DOMAIN}

DNS.2 = www.${ENV::DOMAIN}

DNS.3 = mail.${ENV::DOMAIN}

DNS.4 = ssl.${ENV::DOMAIN}

DNS.5 = localhost

DNS.6 = localhost.localdomain

IP.1 = 127.0.0.1

IP.2 = ::1

bash script for initialising the CA. Must be executed in an empty directory. Before execution, ca.cfg must be copied into it.

**Attention! Everything in the directory will be erased!**

#!/bin/bash

rm index.\*

rm -rf private

rm -rf newcerts

rm -rf certs

rm -rf csr

rm -rf crl

rm crlnumber

rm crlnumber.\*

rm serial

rm serial.\*

mkdir -p private

mkdir newcerts

mkdir certs

mkdir csr

touch index.txt

echo "01" > serial

echo "01" > crlnumber

mkdir crl

export DOMAIN=example.com

export EMAIL=test@example.com

export ADM="Physician"

export ADMAUTH="Medical Chamber Test Country"

export TID="1-109900000000000099"

echo "#####################Generate Root #################"

openssl genrsa -aes256 -out private/cakey.pem 4096

openssl req -config ca.cfg -key private/cakey.pem -new -x509 -days 3650 -sha256 -extensions v3\_ca -out certs/cacert.pem

echo "####################Generate OCSPSigner ##################"

openssl genrsa -out private/ocsp.key.pem 3072

openssl req -config ca.cfg -subj "/C=DE/O=eHealth OTH-Regensburg/CN=Test OCSPSigner 1:PN" -key private/ocsp.key.pem -new -sha256 -out csr/ocsp.csr.pem

openssl ca -config ca.cfg -extensions ocsp -days 700 -notext -md sha256 -in csr/ocsp.csr.pem -out certs/ocsp.cert.pem

echo "###################Generate first CRL#################"

openssl ca -config ca.cfg -gencrl -out crl/ca.crl.pem

openssl crl -in crl/ca.crl.pem -text

echo "#####################Generate Server (TLS) Zertifikat ######################"

openssl genrsa -out private/tlssigner.key.pem 3072

openssl req -config ca.cfg -key private/tlssigner.key.pem -new -sha256 -out csr/tlssigner.csr.pem

openssl ca -config ca.cfg -extensions server -days 700 -notext -md sha256 -in csr/tlssigner.csr.pem -out certs/tlssigner.cert.pem

bash script for the fast generation of user-auth certificates

In productive operation, please query TID (Doctor’s Number) via read

#!/bin/bash

./openssl version -a

echo "No. of certificate file: "

read i

echo "First Name: "

read VORNAME

echo "Last Name: "

read NACHNAME

echo "serialNumber in DN: "

read SERIALNUMBER

echo "E-Mail: "

read EMAIL

export EMAIL

#export EMAIL="test@example.com"

export ADM="Physician"

export ADMAUTH="Medical Chamber Test Country"

export TID="1-109900000000000099"

#echo "Admission Authority: "

#read ADMAUT

#export ADMAUTH=$ADMAUT

./openssl genrsa -out private/user$i.key.pem 3072

./openssl req -config ca.cfg -multivalue-rdn -subj "/C=DE/serialNumber=$SERIALNUMBER+GN=$VORNAME+SN=$NACHNAME+CN=$VORNAME $NACHNAME" -key private/user$i.key.pem -new -sha256 -out csr/user$i.csr.pem

./openssl ca -utf8 -verbose -config ca.cfg -preserveDN -extensions userauth -days 700 -md sha256 -in csr/user$i.csr.pem -out certs/user$i.cert.pem

./openssl x509 -inform PEM -in certs/user$i.cert.pem -outform DER -out certs/user$i.crt

./openssl pkcs12 -export -in certs/user$i.cert.pem -inkey private/user$i.key.pem -out private/user$i.p12

./openssl asn1parse -inform DER -in certs/user$i.crt -dump -i

Bash script for TLS certificates

#!/bin/bash

echo "No. of certificate file: "

read i

echo "Organisation: "

read ORGANISATION

echo "Common Name: "

read CN

echo "DOMAIN: "

read DOMAIN

echo "E-Mail: "

read EMAIL

export DOMAIN

export EMAIL

#export EMAIL="test@example.com"

export ADM="Physician"

export ADMAUTH="Medical Chamber Test Country"

export TID="1-109900000000000099"

#echo "Admission Authority: "

#read ADMAUT

#export ADMAUTH=$ADMAUT

#./openssl genrsa -out private/user$i.key.pem 3072

#./openssl req -config ca.cfg -multivalue-rdn -subj "/C=DE/serialNumber=$SERIALNUMBER+GN=$VORNAME+SN=$NACHNAME+CN=$VORNAME $NACHNAME" -key private/user$i.key.pem -new -sha256 -out csr/user$i.csr.pem

#./openssl ca -utf8 -verbose -config ca.cfg -preserveDN -extensions userauth -days 700 -md sha256 -in csr/user$i.csr.pem -out certs/user$i.cert.pem

#./openssl pkcs12 -export -in certs/user$i.cert.pem -inkey private/user$i.key.pem -out private/user$i.p12

#./openssl asn1parse -inform DER -in certs/user$i.crt -dump -i

echo "#####################Generate Server (TLS) Zertifikat ######################"

openssl genrsa -out private/tls$i.key.pem 3072

openssl req -config ca.cfg -subj "/C=DE/O=$ORGANISATION/CN=$CN" -key private/tls$i.key.pem -new -sha256 -out csr/tls$i.csr.pem

openssl ca -utf8 -verbose -config ca.cfg -extensions server -days 700 -notext -md sha256 -in csr/tls$i.csr.pem -out certs/tls$i.cert.pem

openssl x509 -inform PEM -in certs/tls$i.cert.pem -outform DER -out certs/tls$i.crt

openssl asn1parse -inform DER -in certs/tls$i.crt -dump -i