Secure and Dependable Systems
Jacobs University Bremen
Zihan Qi

Course: CO21-320203
May 16th, 2018

Homework 5

Problem 5.1 *Pretty Good Privacy (PGP)*

Solution:
(a) Matriculation number: 30000612
30000612 + 663890472 = 693891084

(b) A key revocation certificate is a special, revoked copy of your public key. You can generate a key revocation certificate and store it for future use. Key revocation certificates are especially useful if you've forgotten the passphrase to your private key and you need some way to "disable" or revoke that key. Since you've forgotten the passphrase or lost the private key, the only way to revoke the key will be with a revocation certificate that you generated earlier (when you still remembered the passphrase and had the private key). In a way, a key revocation certificate is a kind of insurance plan that lets you keep ultimate control over your key, even if you lose the private key or forget the passphrase.
(Reference: http://www.spywarewarrior.com/uiuc/ss/revoke/pgp-revoke.htm)

Problem 5.2 *X.509 Certificates*

Solution:
(a) Generate a RSA public/private key pair:
```
$ openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:2048
```

Size used: 2048 bits
Extract the public key into a separate file:
```
$ openssl rsa -pubout -in private_key.pem -out public_key.pem
```
This command extract public_key file to current working directory.

(b) Generate a Certificate Signing Request (CSR) for the RSA public/ private key pair:(-sha256 is optional)
```
$ openssl req -new -sha256 -inform PEM -in private_key.pem -out private_key.csr
```

Show the content of the CSR:
```
$ openssl req -text -noout -verify -in private_key.csr
```

(c) Generate a self signed certificate for the CA:
OpenSSL uses the information you specify to compile a X.509 certificate using the information prompted to the user, the public key that is extracted from the specified private key which is also used to generate the signature.

First, we generate a key:
```
$ openssl genrsa -out ca.key 2048
```
Then generate a self signed certificate for the CA:
```
$ openssl req -new -x509 -key ca.key -out ca.crt
```

(d) Sign a CSR with CA:
```
$ openssl x509 -req -in private_key.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out out.crt
```

Problem 5.3 *TLS and Certificate Validation*

Solution:
(a) The entire HTML document that that URL holds is returned. If execute $ curl -v URL, the information also includes TLS handshake procedure and Server Certificate and more info.
(b) By comparison, for www.jacobs-university.de and cnds.eecs.jacobs-university.de, the differences are:
HTTP Server Header: nginx ; Apache/2.4.25 (Debian)
SSL certificate key length: 4096 ; 2048
Also, except obvious differences like Subject names, cnds.eecs.jacobs-university.de support TLS 1.2, TLS 1.1, TLS 1.0, where the other one only support TLS 1.2, TLS 1.1 protocols. Also, the ciphers supported by the server are different.