SINGAPORE UNIVERSITY OF
TECHNOLOGY AND DESIGN

Established in collaboration with MIT

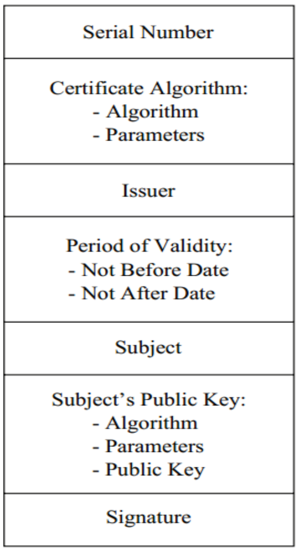
ISTD 50.005 (Computer System Engineering)
Generate RSA certificate request

1 Objective

This supplementary document will show you how to generate an RSA certificate request.

2 Background

In public cryptography, we can freely distribute public keys. However, in reality, we need to ensure the **authenticity** of the public key — we need to know that a certain public key really belongs to certain person. This issue is often solved using a **certificate**, which binds a public key to a known entity. The figure below illustrates the fields of an X.509 certificate, which is widely used for authentication over the Internet.



Serial Number

Certificate Algorithm:
- Algorithm
- Parameters

Issuer

Period of Validity:
- Not Before Date
- Not After Date

Subject

Subject's Public Key:
- Algorithm
- Parameters
- Public Key

Signature

Figure 1. Fields of an X.509 certificate

The authenticity of a public key relies on a certificate. However, one issue still remains: how can we assure the authenticity of the certificate? This issue can be solved as follows: you create a certificate request that contains your public key and personal information, then ask a well-known provider to sign your certificate request. The result of this is a signed certificate. Since we can trust the provider that signs the certificate, we can be sure of the identity of the entity holding the public key and certificate. In reality, well-known providers may be a company like VeriSign or a government authority like IDA. In this assignment, the CSE teaching stuff will serve as your trusted CA.

3 Generate certificate request

You will use OpenSSL to generate your RSA key pair and create a certificate request.

Here are the steps to follow:

1. **Generate RSA keypair**

   ```
   openssl genrsa -out privateServer.pem  1024
   ```

   1024 in the above command means that the modulus of RSA key is 1024-bit length.

2. **Generate certificate request**

   ```
   openssl req -new -key privateServer.pem -out server.csr
   ```

Keys created by OpenSSL can be used in Java if you store them in `.der` format. To convert `.pem` to `der` format, do as follows:

1. **Save the private key in `.der` format**

   ```
   openssl pkcs8 -topk8 -in privateServer.pem -outform der
   -out privateServer.der –nocrypt
   ```

   The `-nocrypt` option is used to output an unencrypted private key that can be used with the Java Cryptography Extension (JCE).

2. **Export public key in `.der` format**

   ```
   openssl rsa -in privateServer.pem -pubout -outform der -
   out publicServer.der
   ```

4 What you have to do?

Generate an RSA key pair and save them in `.der` format for use in your NS project.

You should also generate a certificate request for that keypair and upload the `.csr` file to eDimension.

After you have submitted your certificate request, the CSE teaching staff will sign it and notify you when you can download the signed certificate.