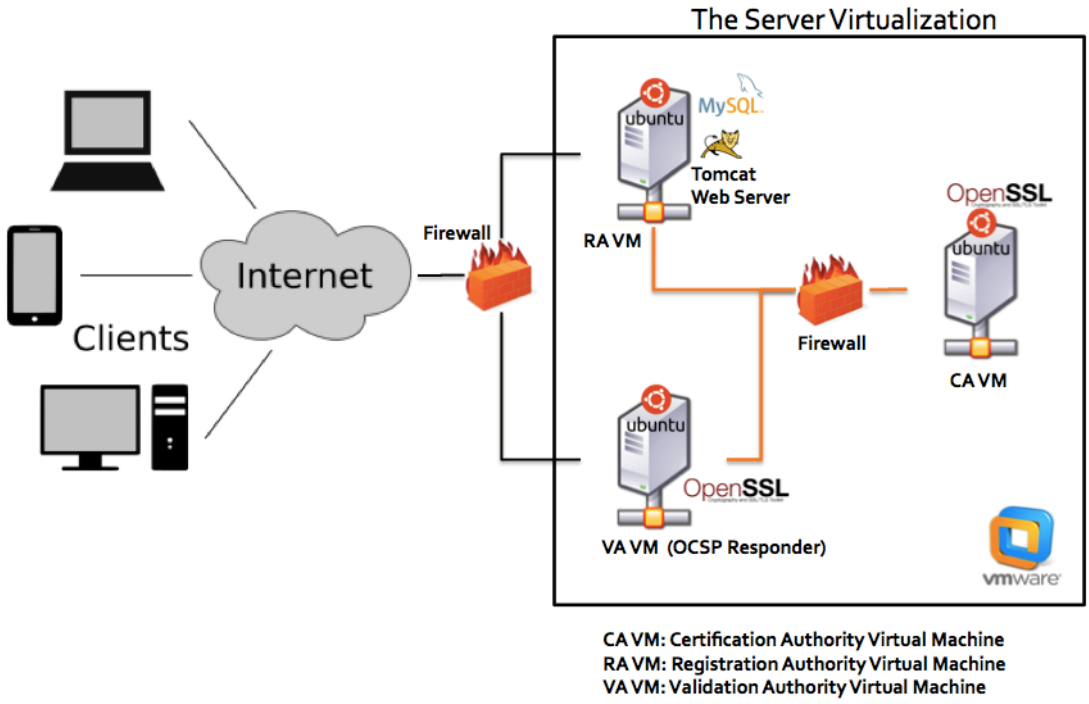
The User Guide of TIHM PKI System

Edited By

Dr. Salaheddin Darwish
(Salaheddin.darwish@rhul.ac.uk)

Table of Contents

1. TIHM PKI Infrastructure Introduction



The primary aim of the PKI system is to create digital certificates for the entities involved in the project. These certificates help establish a secure connection between communicating parties using protocols such as TLS1.2. Additionally, the

system has some features that enable the confirmation of the current certificate's validity when necessary. The PKI infrastructure comprises three essential services: registration (RA), validation (VA), and certification (CA).

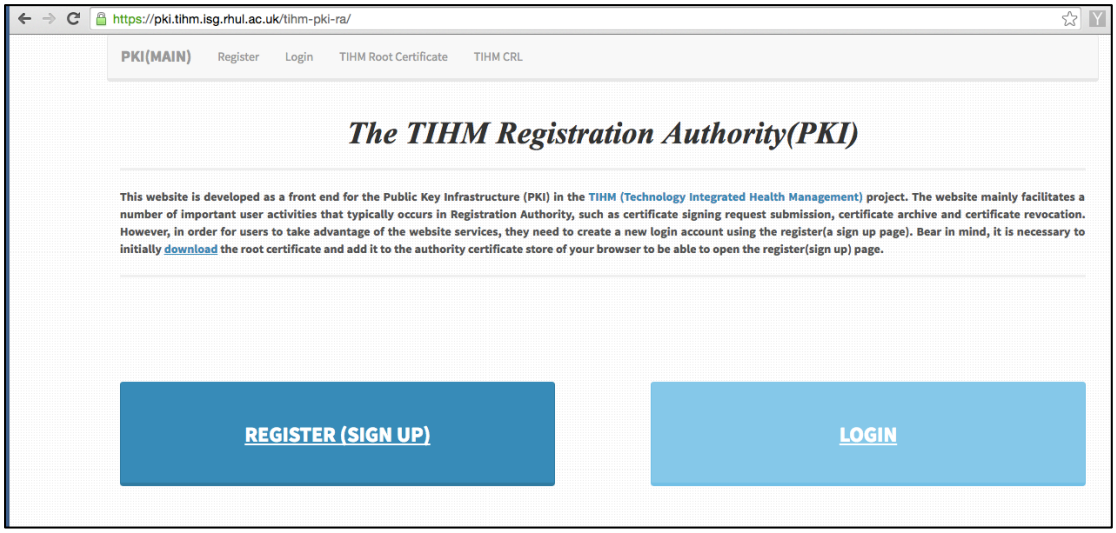
Only registration (RA VM) and validation (AV VM) authority services are accessible to external users. However, the TIHM certification authority (CA VM), which contains the master key for signing certificates, is proposed to be well protected and is anticipated not to be online for use at all times but only in certain time slots for security purposes. The Following URLs for registration and validation services show how to reach these services respectively:

1- http://pki.tihm.isg.rhul.ac.uk [Unavailable Due to Project Completion]
2- http://ocsp.tihm.isg.rhul.ac.uk [Unavailable Due to Project Completion]

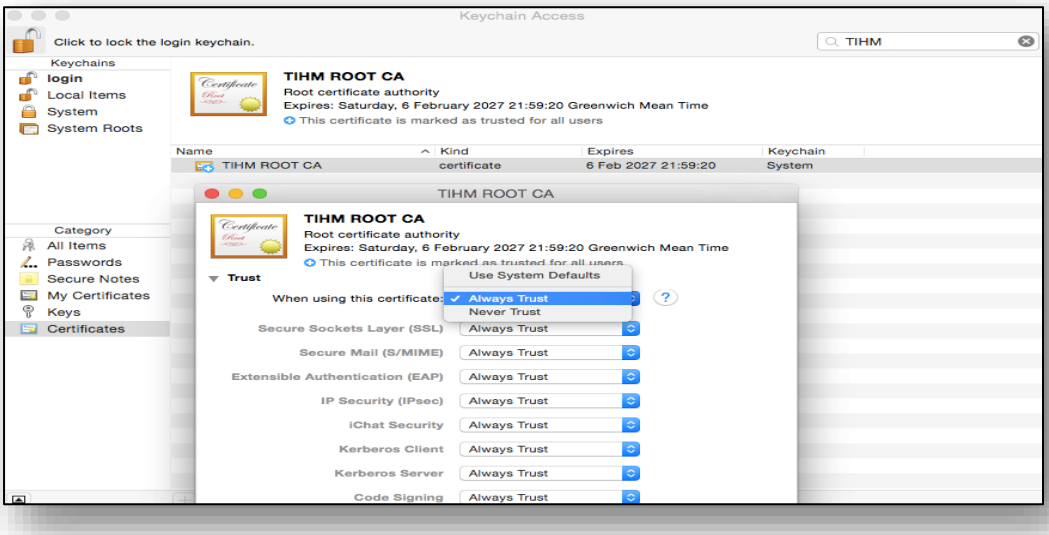
Bear in mind, the URL of the validation service is already included in every certificate being issued by CA. Please check the Authority Information attributes in the certificate.

2. Registration & Login

To create an account, you need first to download the TIHM root certificate and then add it to the authority certificate store of the browser in use.

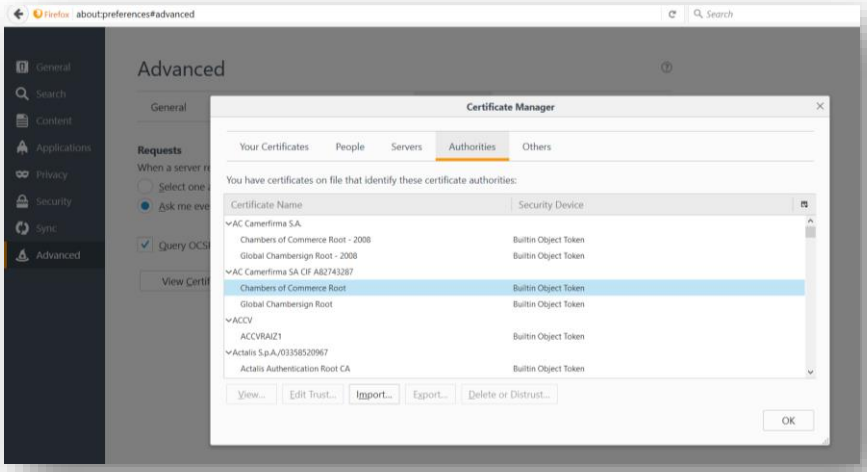


- The examples of feeding the root certificate into different browsers
  - Adding a certificate in Mac for Chrome, Firefox and Safari:

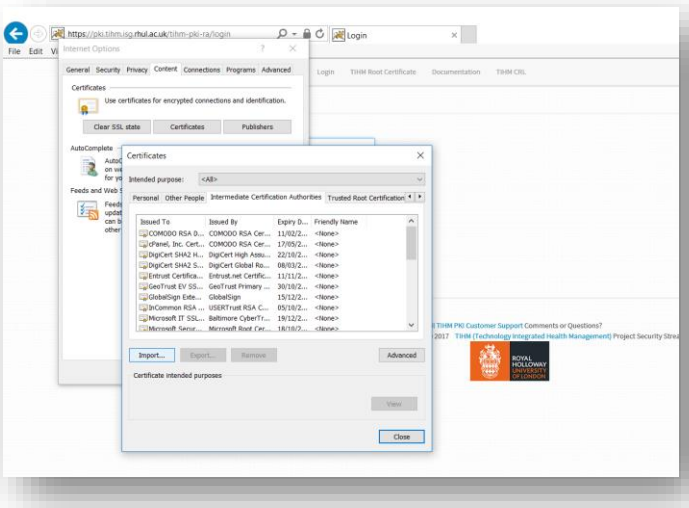


- Adding a certificate in Windows:

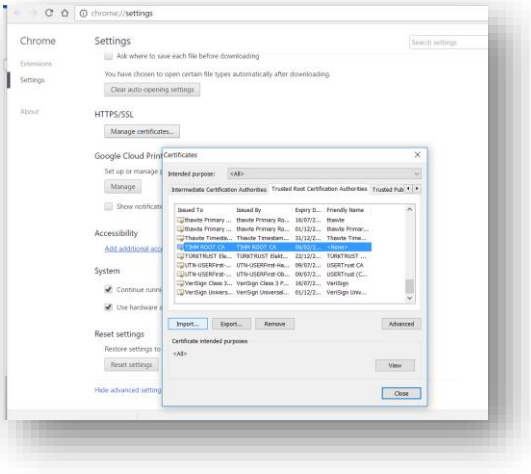
- For Firefox browser:



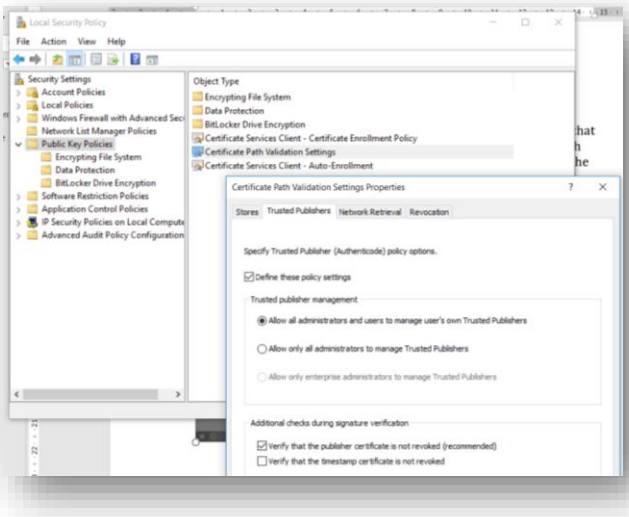
- For Internet Explorer and Edge browsers:-



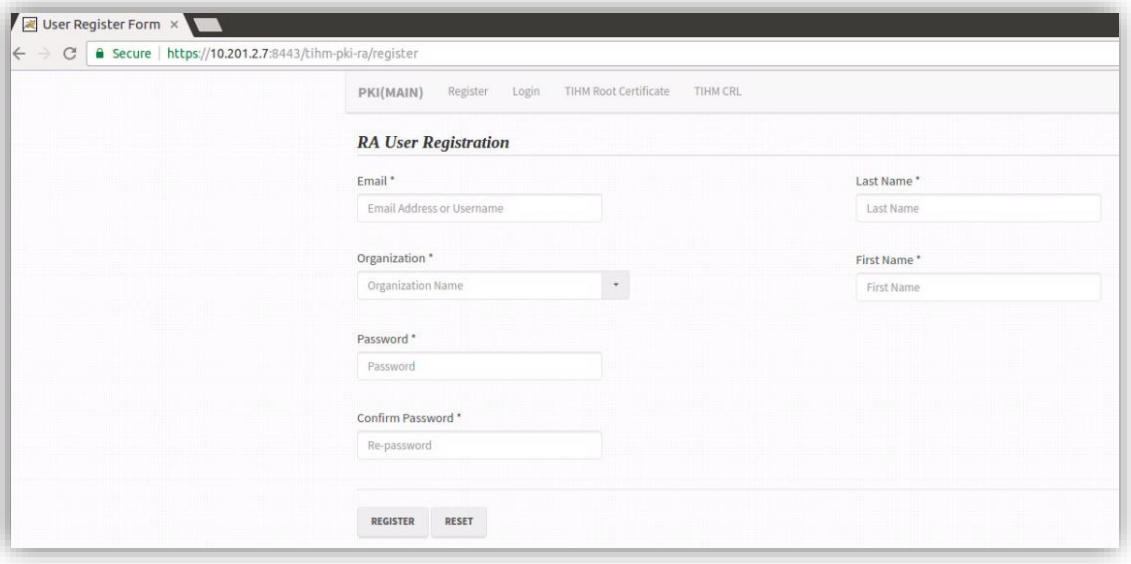
- For Chrome browsers:-



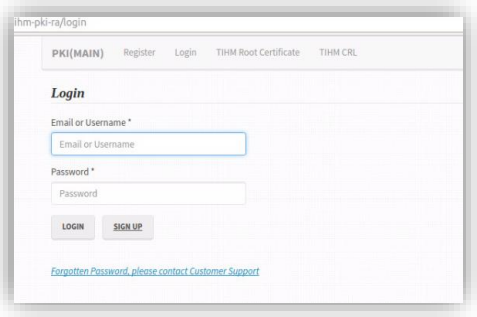
Important Note for revocation: it is important to enable the revocation check in every browser and each browser has a different way of configuring that feature. For example, Firefox has an option for OCSP checking which you need to activate. Windows Internet Explorer has some checks in the advanced settings, which you have to enable in addition to some options in the local security policy. See the next screenshot:



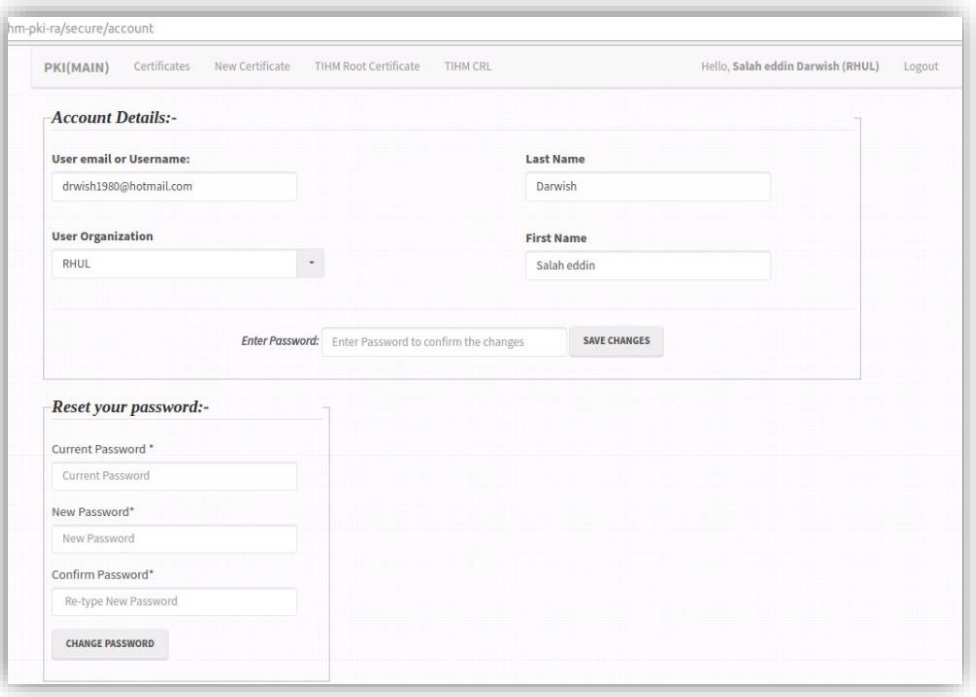
Filling out the registration form for a user to login to the system, therefore you should provide a username, password and details.



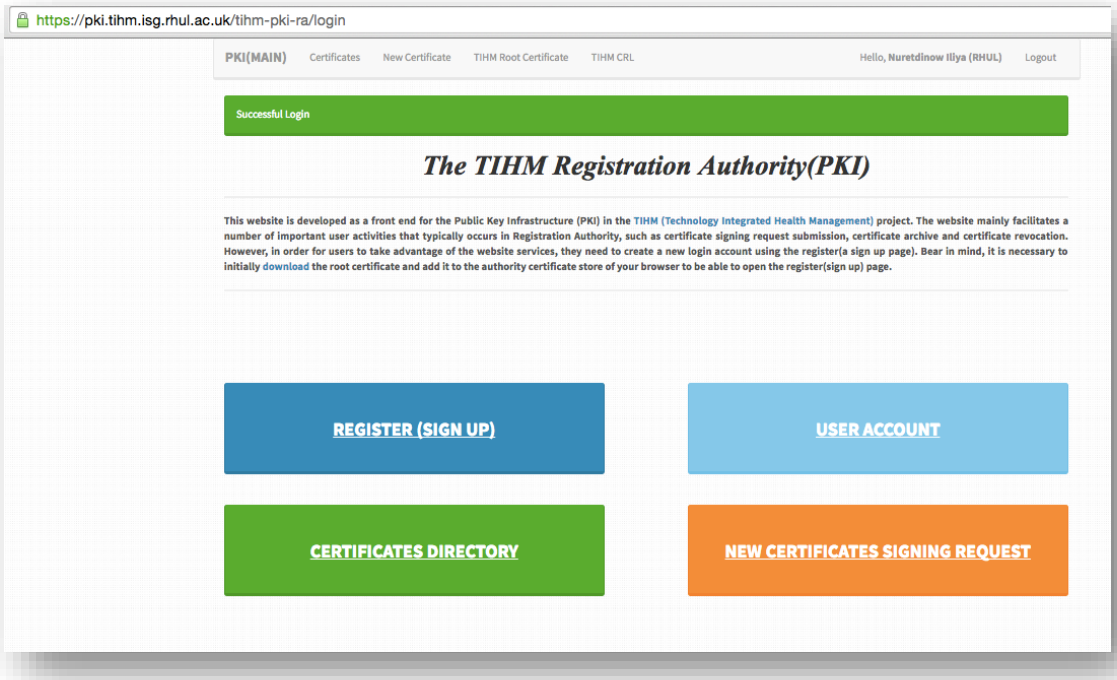
You can access the system by using Login page



The next page is to allow users to manage their login accounts, such as making changes to user details or change the password.

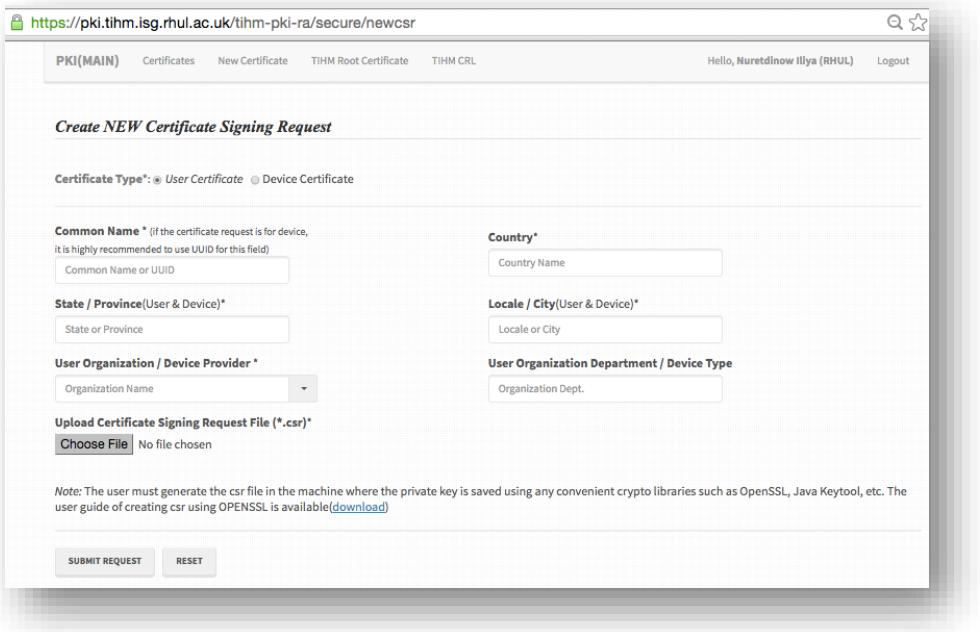


The following screen is available after logging in; the main page will include four buttons referring to different system activities:



3. Submitting a Certificate Singing Request

The next page allows user to feed the new certificate-signing request to the system. Firstly, before you fill out the form on the page, you need to generate a csr file any convenient security tools. Bear in mind, that the organization name that appears in the request must match the user organization inserted during the user registration.



Example: How to generate the CSR file using OPENSSL in Linux: The following steps are to demonstrate how to create a Certificate Signing Request (*.csr) file using a Linux terminal, so you can upload to RA to generate a request.
Generating the CSR requires a number of shell commands, the location and file name of your newly created key, and a path and file name for your CSR. In addition, some information (e.g. country, state, etc.) is required to populate the CSR.

1) Generate a pair of keys (public and private keys) and save file in path with key file extension, it is important to provide the size of keys in this example 2048bit:

   openssl genrsa -des3 -out /path/to/www_server_com.key **2048**

2) Then, create a configuration file (*.txt file) which includes all the information and properties required for generate a correct certificate request for CA. in

   cat > csr_details.txt <<-EOF

   [req]
   default_bits = 2048
   prompt = no
   default_md = sha256
   req_extensions = req_ext
   distinguished_name = dn

   [dn]
   C=GB # The **Country** Name is mandatory and takes a two-letter country code such as GB.
   ST=Surrey # The **State or Province Name** field requires a full name – do not use an abbreviation, such as London.
   L=Egham # The **Locality Name** field is for your city or town or device address.
   O= RHUL # The **Organization Name** field, insert your organization or device provider
   OU=ISG # The **Organizational Unit** Name is an optional field for your department or section or device type
   CN = www.your-new-domain.com # The Common Name field is used as a user or device identifier. For a user identifier, enter your name while for a device identifier, it is recommended to use UUDI to have a unique identifier.
   emailAddress=your-administrative-address@your- domain.com # An optional filed

   [req_ext]
   subjectAltName = @alt_names

   [alt_names]
   DNS.1 = your-new-domain.com # alternative Name for CN
   DNS.2 = www.your-new-domain.com # alternative Name for CN
   EOF
   After performing the previous command, please check if the configuration file has been created by looking into the current directory.

3) Finally, create Certificate Signing Request (*.csr) file by using the following OpenSSL command:

   openssl req -new -key -out /path/to/www.your-new-domain.com.csr **-key** /path/to/www_server_com.key **-config** <(**cat** csr_details.txt)

Note that a certificate-signing request always should have a file name ending with a *csr* file extension. During the execution of the previous command, entering your passphrase will be prompted for the key.

Note: If you prefer to use other tools to generate your own csr file, the following links provide you with some relevant resources of the other tools such as java keytool, windows MMC , etc.:

- **Java Keytool:**

  https://www.digicert.com/csr-creation-java.htm

- **Windows:**

  http://www.entrust.net/knowledge-base/technote.cfm?tn=8924

  https://documentation.meraki.com/zGeneral_Administration/Other_Topics/Creating_an_Offline_Certificate_Request_in_Windows_Server

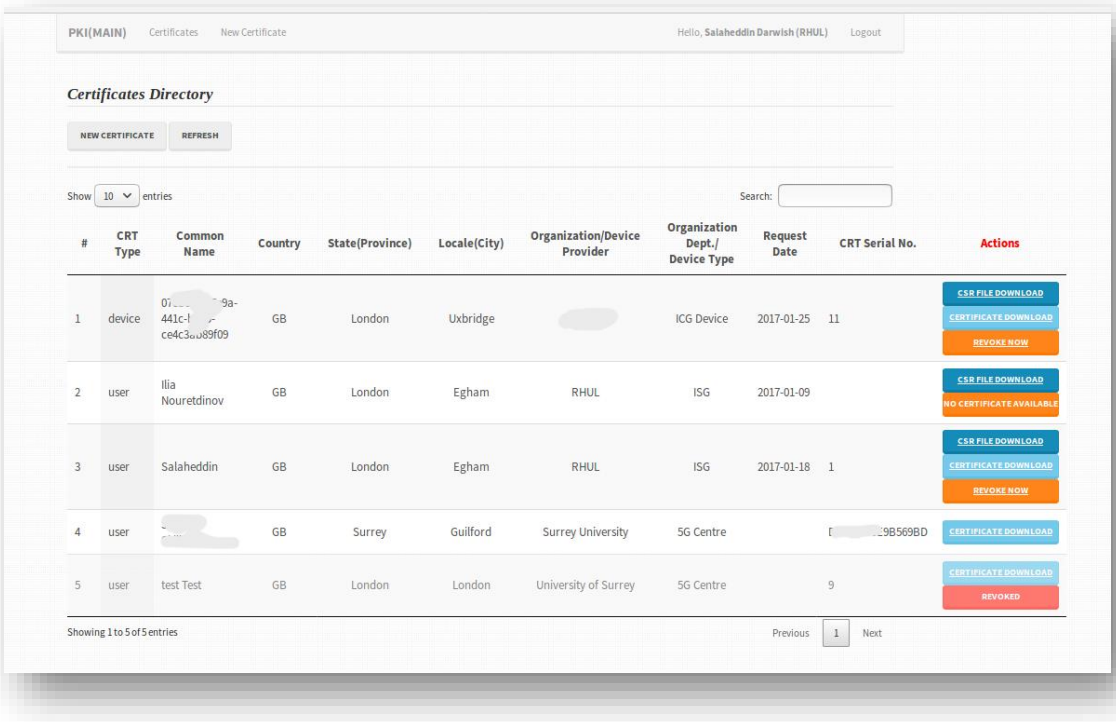
- **Mac:**

  https://www.digicert.com/csr-creation-ssl-installation-mac-osx-el-capitan.htm

  https://www.sslsupportdesk.com/certificate-signing-request-csr-instructions-for-apple-mac-os-x-10-11/

4. Checking Certificates and Revocation

Browse the certificate directory:



After submitting a CSR request, refresh the screen in some time to see when it becomes ready for CERTIFCATE DOWNLOAD. The certificate is expected to take at most 20 minutes. For revocation of a certificate, use REVOKE NOW button. The status will be then changed to REVOKED once the revocation request is confirmed by the CA.

5. OCSP responder (Certificate validation)

There is no GUI for this service (http://ocsp.tihm.isg.rhul.ac.uk[Unavailable Due to Project Completion]) and the service is expected to be incorporated in the TLS protocol between two parties to validate certificates being exchanged before establishing a secure channel. Every entity must configure their ends to call back this service.

Example of OPENSSL command line to call this service

- **openssl ocsp -issuer** ~/va/TIHM-CA.crt **-nonce -CAfile** ~/va/TIHM-CA.crt **-url http://ocsp.tihm.isg.rhul.ac.uk -serial** "0x04" **-resp_text**

6. CRL (Certificate Revocation List)

Apart from the OCSP responder, the PKI system is designed to provide CRL to the users as an alternative when OCSP is not available and this list is accessible by http://pki.tihm.isg.rhul.ac.uk/tihm-pki-ra/crl[Unavailable Due to Project Completion]