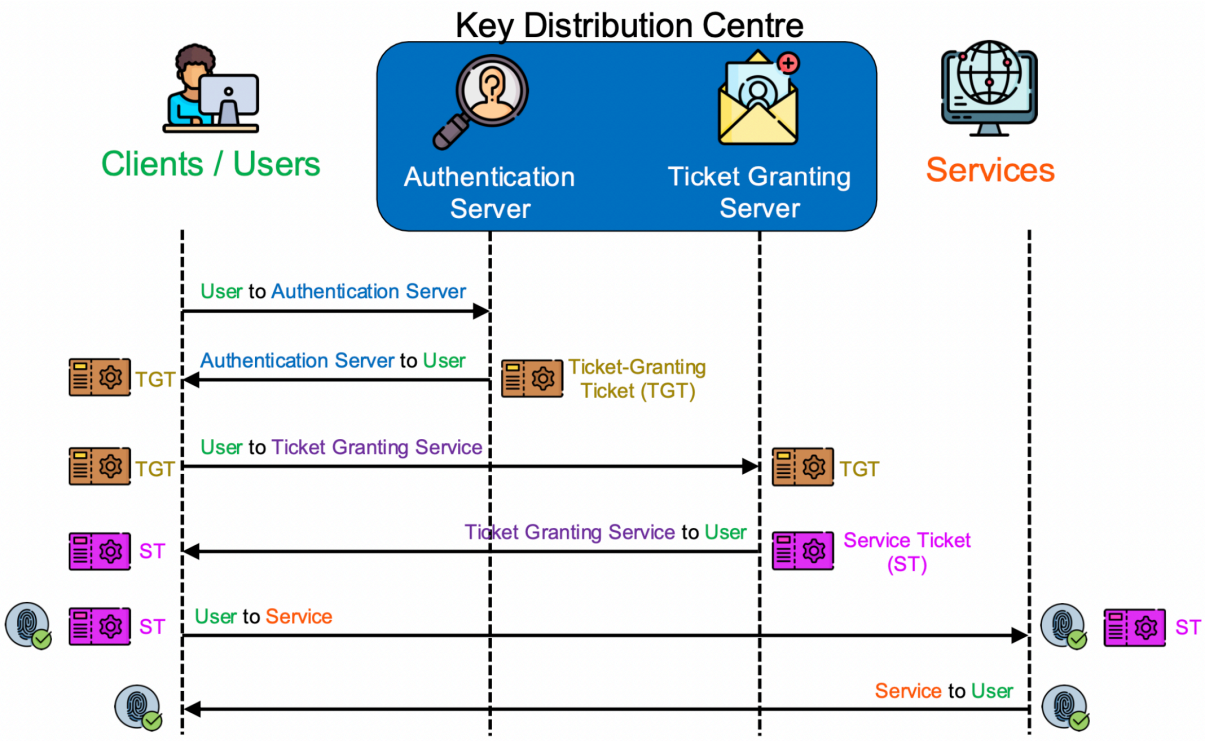
The University of Newcastle
School of Information and Physical Sciences

SENG2250/6250 System and Network Security
Week 9 Lab

Part 1: Distributed System Security

Kerberos is a robust and widely used authentication protocol designed to ensure secure communication in computer networks. Developed at MIT, it provides a trusted method for verifying the identities of users and services, preventing unauthorized access and eavesdropping.



Kerberos uses a system of tickets and encryption to facilitate secure authentication, allowing users to access network resources without exposing their passwords. It has become a cornerstone of network security, particularly in enterprise environments, by enabling single sign-on and reliable authentication across distributed systems, making it an essential tool for safeguarding sensitive data and ensuring network integrity.

Q1: What entities constitute a full-service Kerberos environment?

Q2: In the context of Kerberos, what is a realm?

Q3: Describe the message flow of Kerberos protocol version 4.

Q4: What are the principal differences between version 4 and version 5 Kerberos?

Q5: What are the two tickets generated in (intra-realm) Kerberos protocol version 5? How could they be different in usage? Can we reuse these tickets?

Q6: What are the differences between the intra-realm Kerberos and inter-realm Kerberos protocols?

Part 2: XCA

This lab can be conducted on the Windows 10 VM or your own computer.

Access your virtual lab here:

https://cybersec-vra04.newcastle.edu.au/

Username: Student

Password: $tud3nt

The VM should have XCA installed. If you are using another computer, you may need to install XCA: https://hohnstaedt.de/xca/index.php/download

Please keep a copy of the generated certificates/keys, as they will be used in future labs (or you can generate them again during future labs).

Task 1: Creating a Certificate Authority (CA) using XCA

In this task, we will first create a CA. The role of CA is to issue public-key certificates to end entities (e.g., our web server). In other words, the CA digitally signs a public key of a web server and embeds it into the server's certificate. The CA uses its private key for signing. In practice, the CA certificate manager (a certificate server) should be installed on a secure machine (potentially disconnected from the network); the CA's private key should be kept highly secure.

The CA also issues a self-signed public-key certificate, whereby the CA digitally signs its own public key. This certificate is distributed, in a secure way (the integrity of the CA's certificate must be protected), to all users who use a certificate issued by the CA to verify the authenticity of the certificate holder (e.g., a web server).

1. Open XCA (e.g., C:\Program Files\xca\xca.exe)
   Create a new database to store your keys and certificates. Select "File" and "New DataBase", create a new database named "mydatabase". You will be asked for a password, but it is not necessary for this lab. You can simply click "Ok".

2. Click the "New Certificate" button to start the X.509 certificate creation procedure.

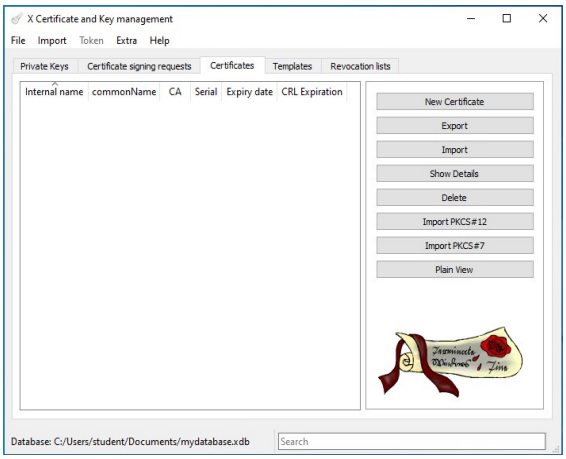


Figure 1. Starting new certificate generation.

3. Select CA template for this certificate.
4. Select the “Subject” tab and fill the fields, i.e., “Internal name”, “Country name”, etc. Please refer to Figure 2 as an example. Click “Generate a new key” button to generate a new private key (2048-bit RSA). This will be the signing key of the CA.
5. Select the “Extensions” tab and set the certificate “Type” to “Certificate Authority”, then click “OK” to finish the certificate creation procedure.

You can check the details of the created CA certificate by double-clicking on it in the main XCA window under the “Certificate” tab.

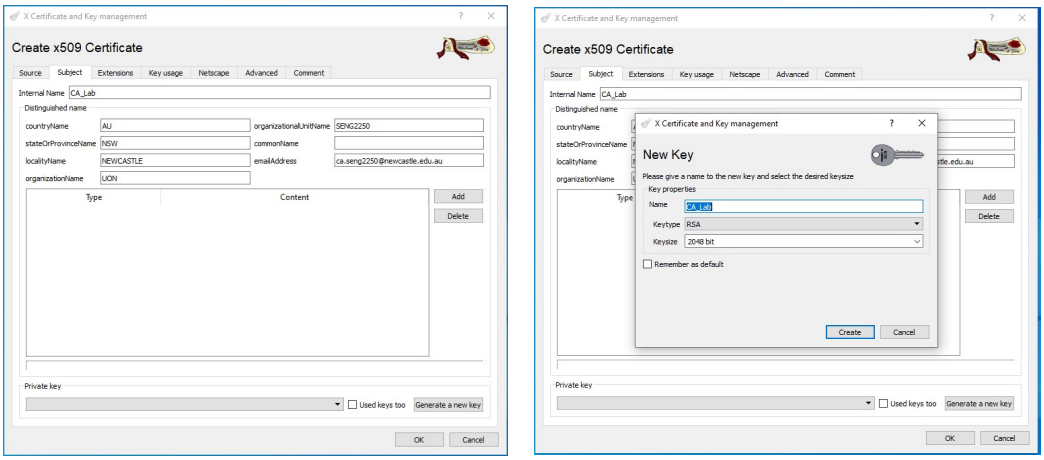


Figure 2. Create a CA certificate.

Task 2: Create a Web Server Certificate

In this task, we create a public-key certificate for a web server. This certificate will be digitally signed by the created Certification Authority (CA).

1. Click the “New Certificate” button in the main XCA window. Select the “Source” tab and configure the web server certificate properties as follows:
    a. Check “Use this Certificate for signing” and set it to the name of the CA created in the previous task.
    b. Choose “TLS_server” (or “HTTPS_server”, if applicable) from the list of available templates.
2. Select the "Subject" tab and fill the fields appropriately. Please refer to Figure 3 as an example. It is important to note that fill the commonName as “localhost”. This will be the IP address (or the corresponding URL) of your web server. Finally, click “Generate a new key” button to generate a new private key (2048-bit RSA).

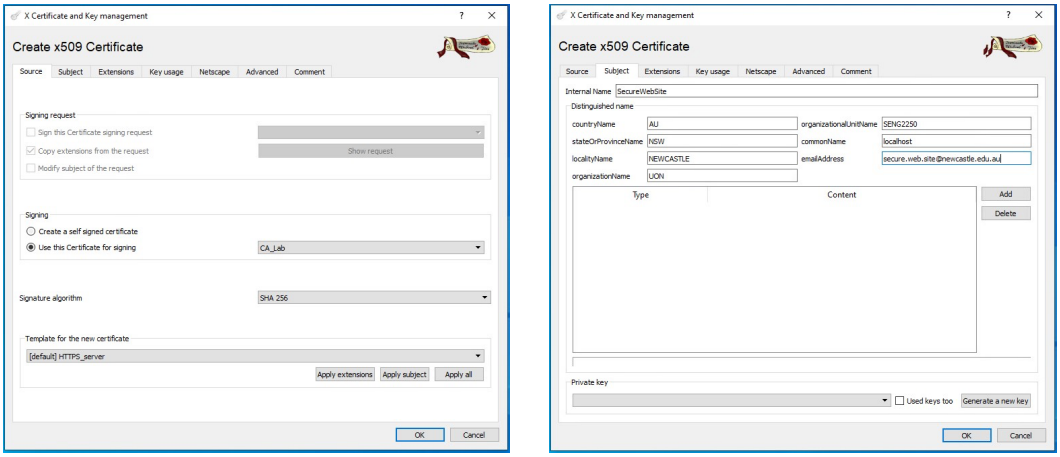


Figure 3. Create web server certificate.

3. Under the “Extensions” tab, set the type to “End Entity”, then click “OK” to finish the certificate creation procedure.

Task 3: Create a Client Certificate

In this task, we create a public-key certificate for a client. The CA will sign this certificate.

1. Click the “New Certificate” button in the main XCA window. Select the “Source” tab and configure the client certificate properties as follows:
    a. Check “Use this Certificate for signing” and set it to the name of the CA created in the previous task.
    b. Choose “TLS_client” (or “HTTPS_client”, if applicable) from the list of available templates.
2. Select the “Subject” tab and fill the fields properly. Please refer to Figure 4 as an example. Then, click “Generate a new key” button to generate a new private key (2048-bit RSA).

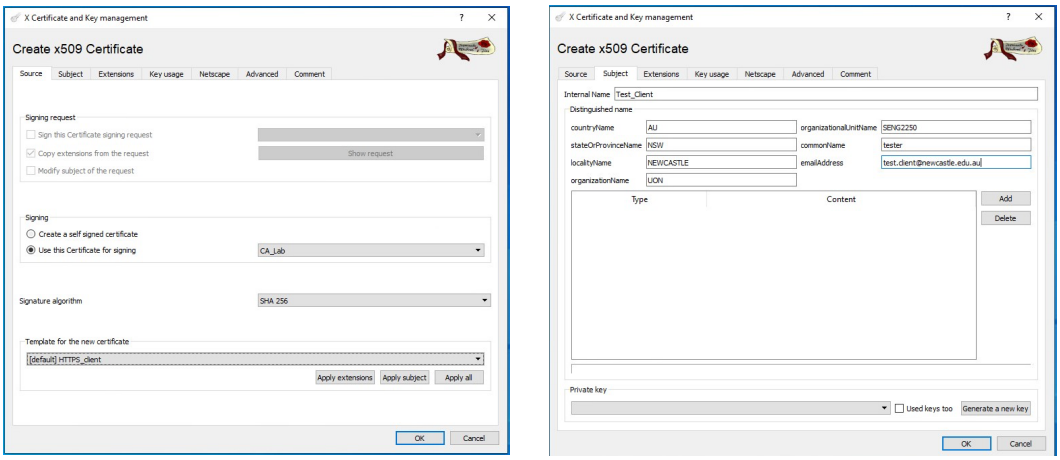


Figure 4. Create client certificate.

3. Under the “Extensions” tab, set the type to “End Entity”, then click “OK” to finish the certificate creation procedure.

Task 4: Export Certificates

In this task, we will export the certificates created in the previous tasks, that is, the CA public-key certificate (without the private key), the web server public-key certificate including its private key, as well as the client public-key certificate with its private key.

1. Open the "Certificates" tab in the main XCA window. Select the CA certificate and click "Export". Select a destination (e.g., Documents) and filename where you want to store the certificate and click "OK".
2. Repeat the above step to export web server and client certificates.
3. Finally, we export the web server and client private keys.
    a. For web server: click the "Private Keys" tab from the XCA main window. Select web server private key and click "Export". Choose a destination (e.g., Documents) and filename where you want to store the private key. Finally, click OK to export the web server private key. (Figure 5)

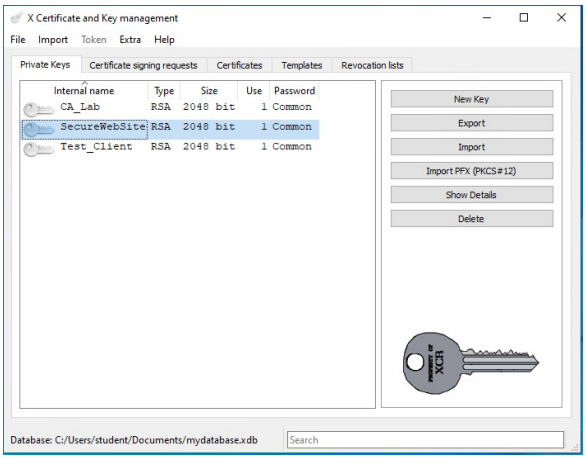


Figure 5. Export web server private key.

    b. For client: select the client certificate under the "Certificates" tab and click "Export". Select a destination (e.g., Documents) and filename where you want to store the private key. Then, change the "Export Format" to "PKCS #12 (.p12)" (Figure 6). It will be imported to the web browser. You will be asked to create a password; you may simply click "OK" to skip it.

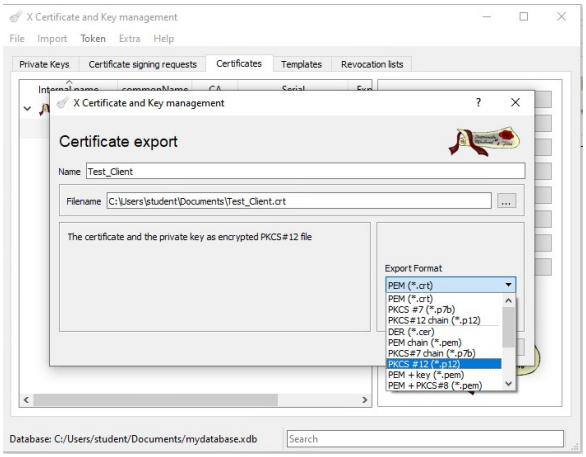


Figure 6. Export client private key.

You should have created files shown as in Figure 7.

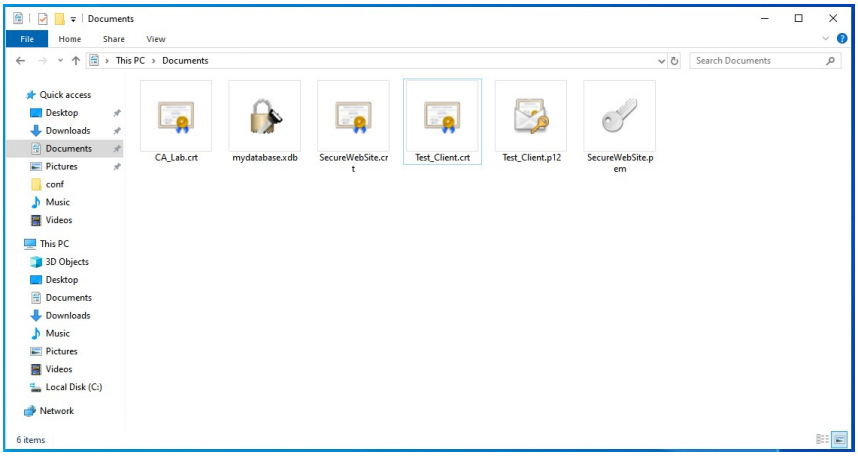


Figure 7. Created files.