**Practical Session 6**

*SSH keys for authentication*

SSH keys enable authentication on modern cloud services and other computer-dependent services They offer convenience and improved security when properly managed.

SSH keys grant access and control who can access what. In identity and access management, they need similar policies, provisioning, and termination as user accounts and passwords. One cannot have confidentiality, integrity, or any guarantees of continued availability of systems without controlling SSH keys.

Technically the keys are asymmetric cryptographic keys that are created using a public key cryptosystem, usually RSA. However, functionally they are authentication credentials and need to be managed as such.

Keys are classified in two. **Authorized keys** are public keys that grant access. **Identity keys** are private keys that an SSH client uses to authenticate itself when logging into an SSH server. Authorized keys and identity keys are jointly called **user keys**. They relate to user authentication, as opposed to host keys that are used for host authentication.

Keys can also being used to implement **certificate-based user authentication** by creating PKI certificates. In this case, the user still has a private key but also has a certificate associated with the key, specifically the public key.

In this practical session, you will create your key pair and a self-signed certificate using the SSH tool on a console. The following instructions are for a linux based console, if you have another operating system, look for the specific instructions, they are very similar.

**Activity**

1. Generating a new SSH key

Open Terminal.

Run the following command by adding your email address. This command indicate SSH to create a RSA 4096 key pair.

$ ssh-keygen -t rsa -b 4096 -C "your\_email@example.com"

This creates a new ssh key, using the provided email as a label. You will see the following message:

> Generating public/private rsa key pair.

When you're prompted to "Enter a file in which to save the key," press Enter. This accepts the default file location.

> Enter a file in which to save the key (/home/you/.ssh/id\_rsa): [Press enter]

At the prompt, type a secure passphrase. For more information, see "Working with SSH key passphrases".

> Enter passphrase (empty for no passphrase): [Type a passphrase]

> Enter same passphrase again: [Type passphrase again]

1. Generating a new certificate.  
    For this, we will use the OpenSSL comand.

openssl x509 -req -sha256 -days 365 -in server.csr -signkey server.key -out server.crt

The server.crt file is your certificate, the server.key is the private key used to sign the certificate, server csr is the file containing the public key that will go with the certificate.

As evidence of this process, copy the corresponding screen shots on this file, explaining the process. Add your conclusions and upload it to Schoology.