Practical Uses of Public and Private Keys Beyond TLS

Yasmine Chaouche (yasmine.chaouche@lecnam.net)
Jacopo Bufalino (jacopo.bufalino@lecnam.net)

November 2025

Public and private key pairs are not only used to verify the authenticity of a web server. There are many different use cases for PKI and public–private key pairs. This lab addresses some of the most common ones.

General Instructions

You will find practical instructions and code for the exercises in Moodle. Each exercise is in a different folder and contains a `README.md` file with further instructions on how to run the exercise. At the end of this document, you will find the submission instructions.

Exercise 1: SSH Key-Based Access

The purpose of this exercise is to configure key-based authentication to a remote SSH server. For this exercise, you will need to use `docker` and `make`. The desktops in the TP room are already equipped with this software.

Follow the steps below:

1. Generate an SSH key pair.
2. Configure key-based authentication.

After completing the exercise, answer the following questions:

1. Does the server know the identity of the client?
2. How does the client know that the server is legitimate?
3. What are the possible vulnerabilities of this authentication mechanism?

Exercise 2: Create Mutual TLS (mTLS) Clients and Server

As part of a finance company's development team, you are in charge of setting up a secure payment system. This system ensures that only authorized customers can communicate with the central server to process payments. To achieve this, the company uses the mutual TLS protocol (mTLS) for client authentication and authorization.

The system is built around a secure payment infrastructure in which the central server, **BankServer**, acts as the central payment processor and ensures that only authorized customers with sufficient funds can process transactions. To establish trust, **BankServer** requires all connecting customers to authenticate themselves using certificates issued by a trusted Certification Authority (CA).

The bank has two registered customers: Alice and Bob. Alice has 1000 €, and Bob has 150 €. At the moment, the server is not secure and relies on HTTP. You are required to:

1. Complete the code in `server.py` to implement mTLS and certificate-based authorization.
2. Create `alice_client.py` and `bob_client.py`, which make requests as Alice and Bob.

Notes on mTLS

mTLS[1][2] is an authentication and authorization protocol based on certificates, in which both clients and servers authenticate each other. It is used in zero-trust infrastructures and corporate environments. You are encouraged to learn more about the topic online.

Exercise 3: Set Up a VPN Server

A software company has offices in two different sites: Site A and Site B. The company also has servers in a cloud location (Cloud Site A). Both sites and the cloud location are in private networks and therefore cannot communicate with each other.

You have been contracted to implement a VPN server in the cloud gateway so that developers in the offices can securely access cloud resources. However, developers in the two locations have different privileges.

Developers at Site A should be granted access to **both the Production and Development servers**, while developers at Site B should **only access the Development server**. For this exercise, you will implement a VPN server using **OpenVPN**[3], a very popular open-source VPN software that is (also) based on PKI (X.509 certificates). The software can be installed via `apk add openvpn`.

You will implement the server on the cloud gateway, while the clients will run on the developers' machines.

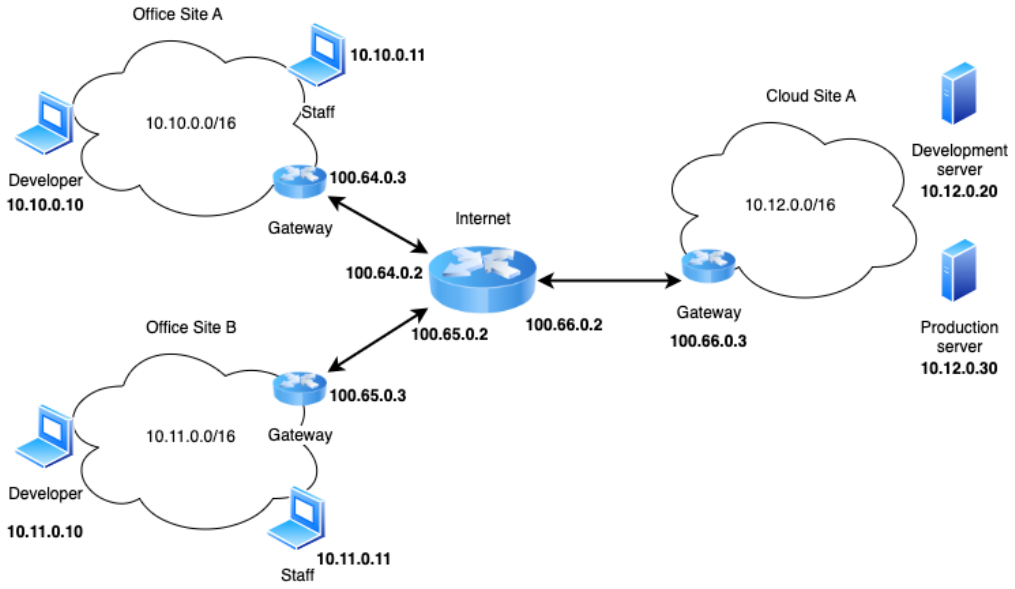

Figure 1: Company network topology

[1] `https://www.cloudflare.com/en-gb/learning/access-management/what-is-mutual-tls/`
[2] `https://cloud.google.com/load-balancing/docs/mtls`
[3] `openvpn.net`

Work Submission Instructions

Please provide a report as a ZIP file, including the following:

1. Exercise 1: Answers to all the questions (in any text format).
2. Exercise 2: The new files, the edited files, and the generated certificates. **The code must run to be evaluated.**
3. Exercise 3: The configuration files and the generated certificates. Also, attach a `.pcap` file captured from the router to demonstrate that packets are encrypted after setting up the VPN.