SSH and SCP

https://github.com/heig-vd-dai-course

Web · PDF

L. Delafontaine and H. Louis, with the help of GitHub Copilot.

Objectives

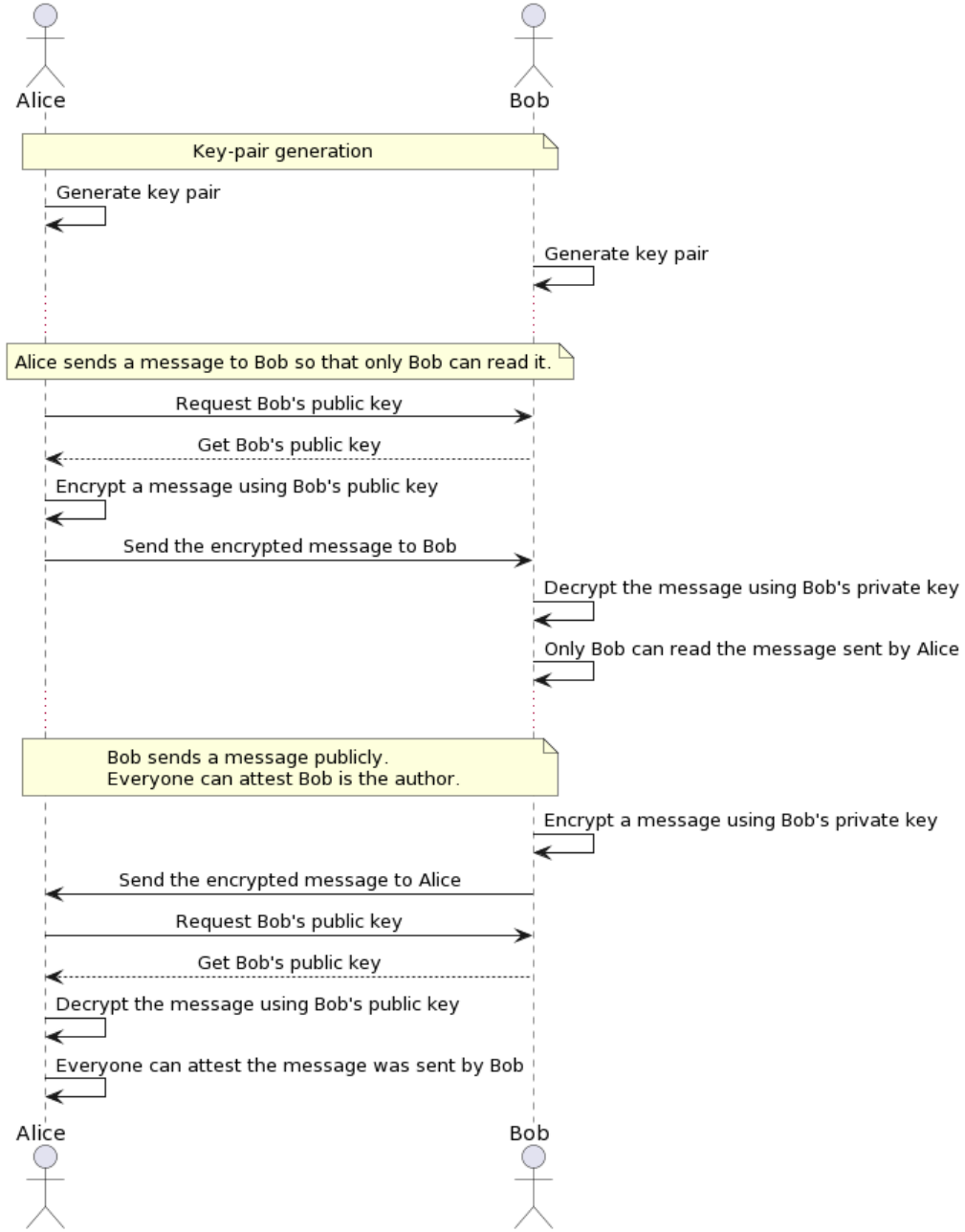
- Refresh on security
- Learn how to use the SSH protocol to connect to a remote server
- Learn how to use the SCP protocol to transfer files to a remote server

A quick reminder about security

More details for this section in the course material. You can find other resources and alternatives as well.

A quick reminder about security

- A secure protocol ensures the confidentiality of the data exchanged
- Most secure protocols rely on cryptography
- Cryptography is based on algorithms and keys

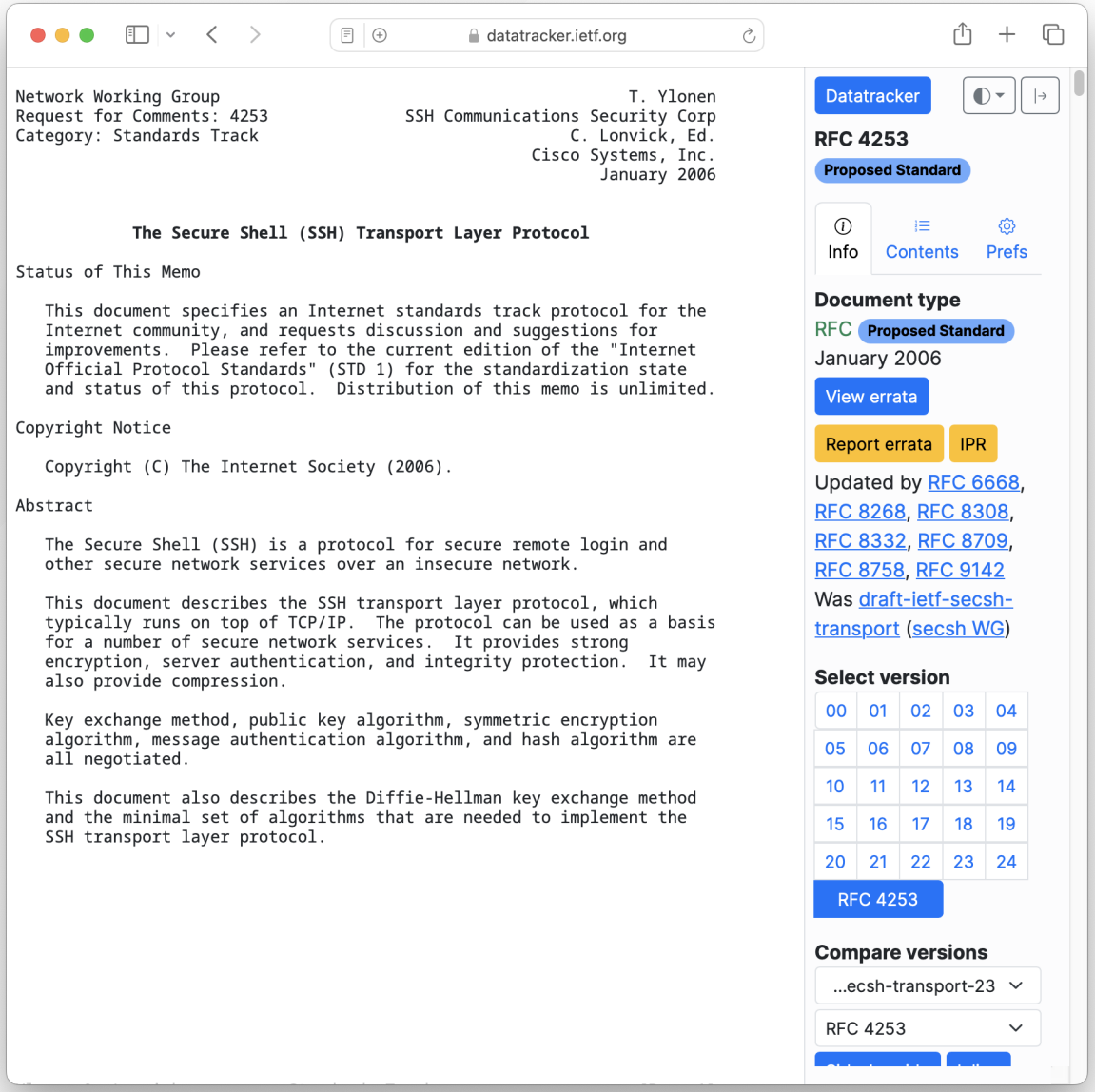


SSH

More details for this section in the course material. You can find other resources and alternatives as well.

SSH

- Secure Shell
- Uses TCP port 22
- A protocol to connect to a remote server
- Can be used to execute commands on a remote server
- The standard way to connect to a remote server

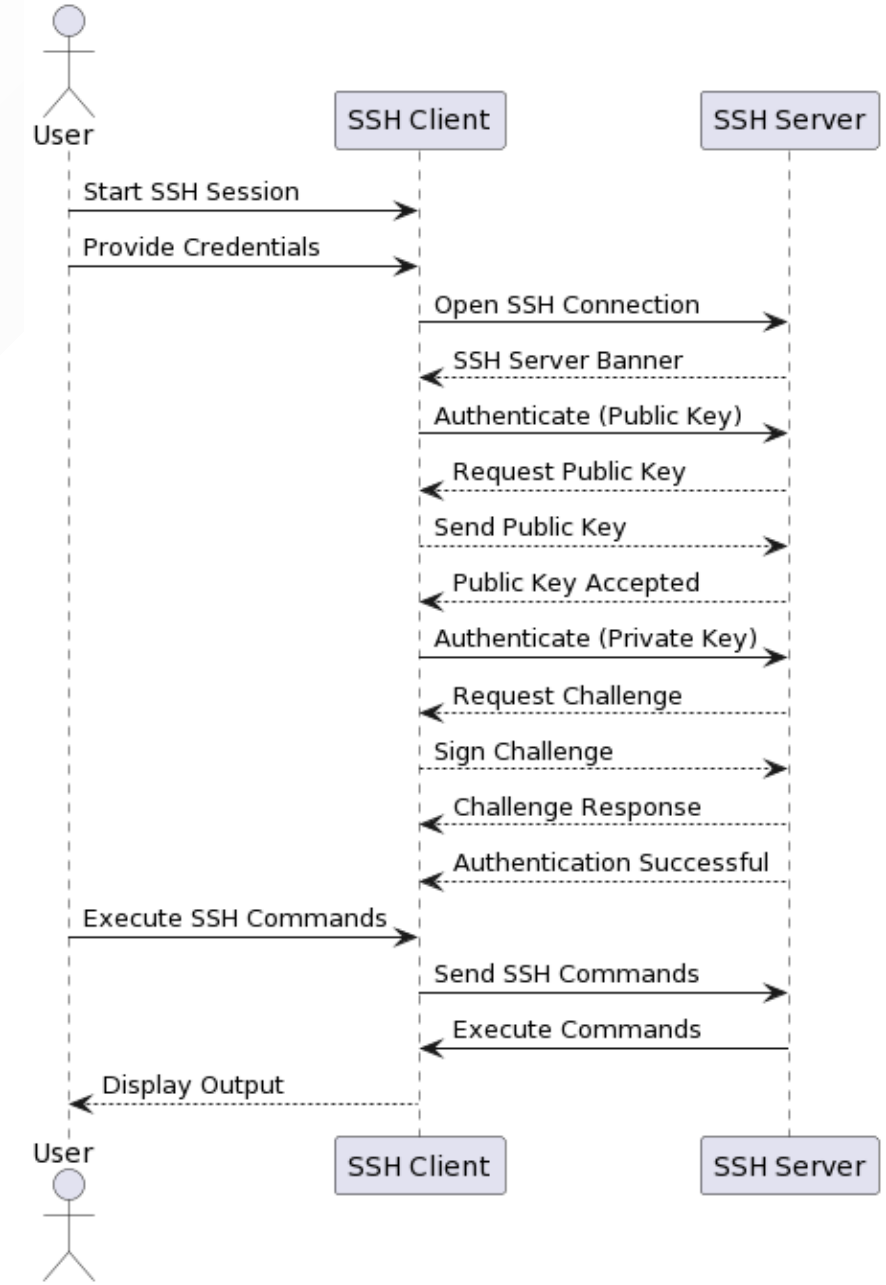


SSH key algorithms

The most common key algorithms are:

- RSA
- DSA
- ECDSA
- Ed25519

Ed25519 and ECDSA are the recommended algorithms.

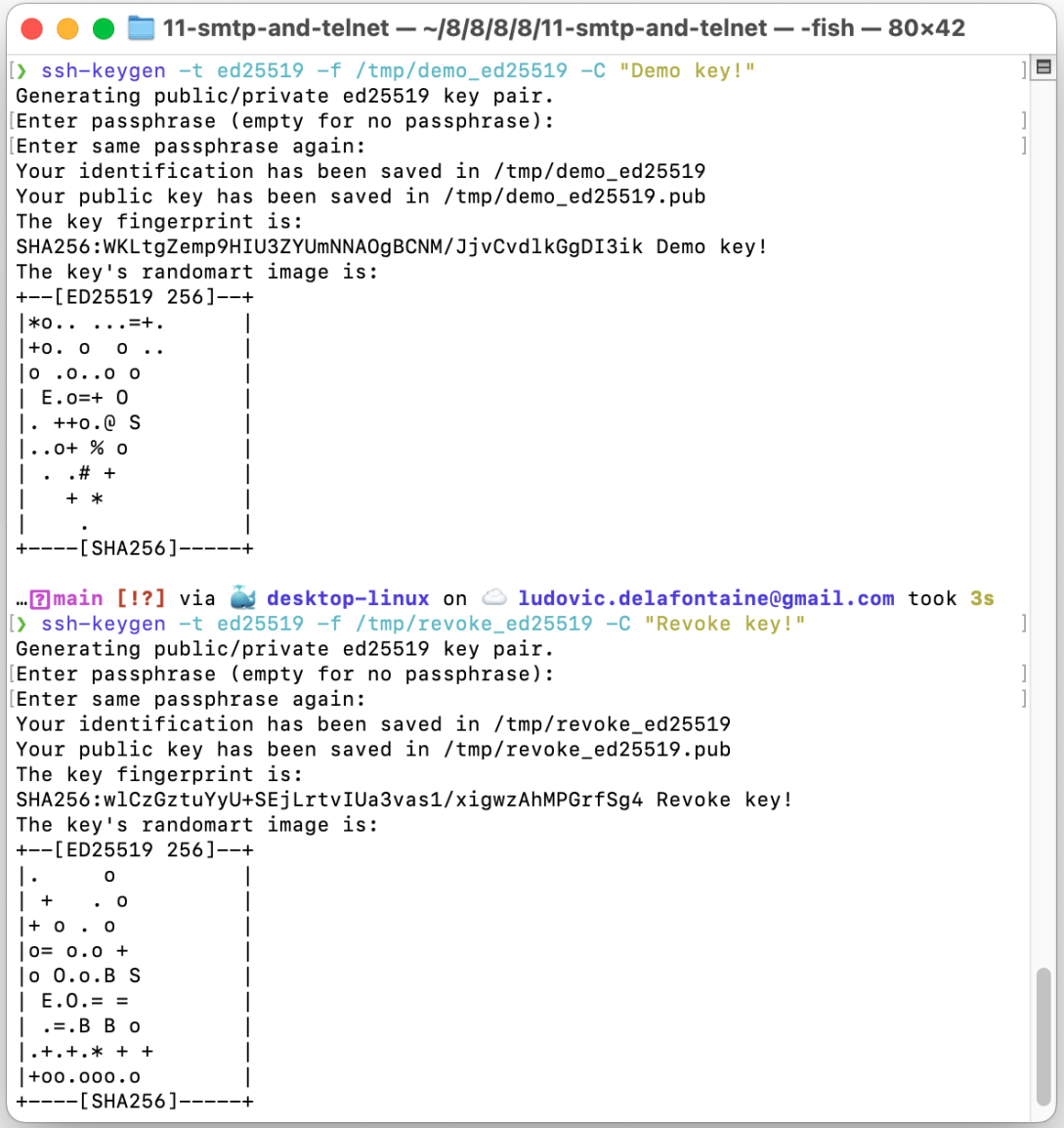


SSH key fingerprint

- Short version of a public key
- Used to verify the identity of a public key
- Can help detect man-in-the-middle attacks
- Stored in the `~/.ssh/known_hosts` file

SSH key generation

- Use the `ssh-keygen` command
- Choose the key algorithm
- Generate a private key and a public key
- Can be done with or without a passphrase

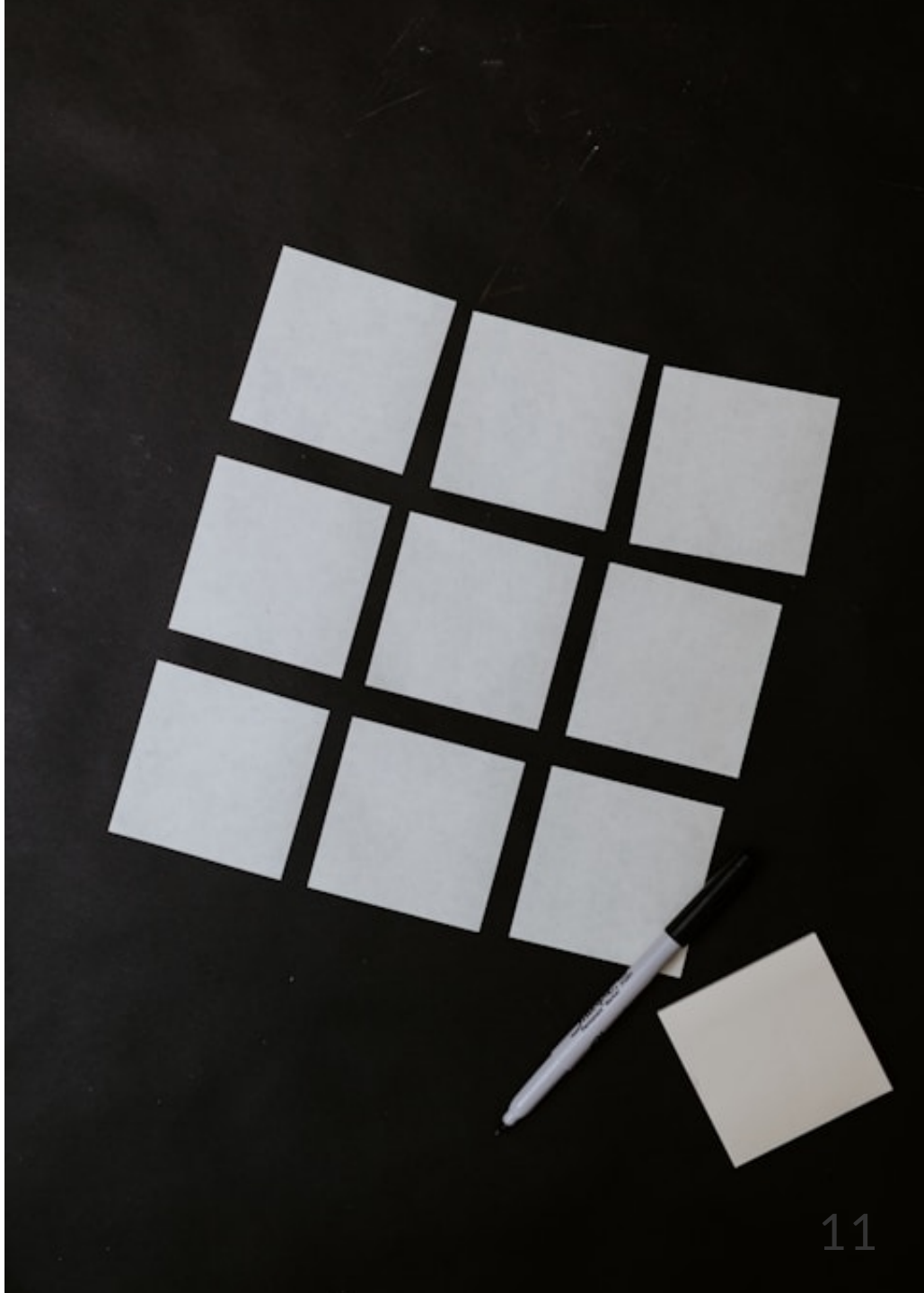


SCP

More details for this section in the course material. You can find other resources and alternatives as well.

SCP

- Secure Copy
- Uses TCP port 22
- A protocol to transfer files to/from a remote server
- Can be used to transfer files between two remote servers as well



Practical content

What will you do?

- Install and configure SSH and SCP
- Start a SSH server with Docker Compose
- Connect to the SSH server with SSH
- Transfer a file to the SSH server with SCP

Find the practical content

You can find the practical content for this chapter on GitHub.

Finished? Was it easy? Was it hard?

Can you let us know what was easy and what was difficult for you during this chapter?

This will help us to improve the course and adapt the content to your needs. If we notice some difficulties, we will come back to you to help you.

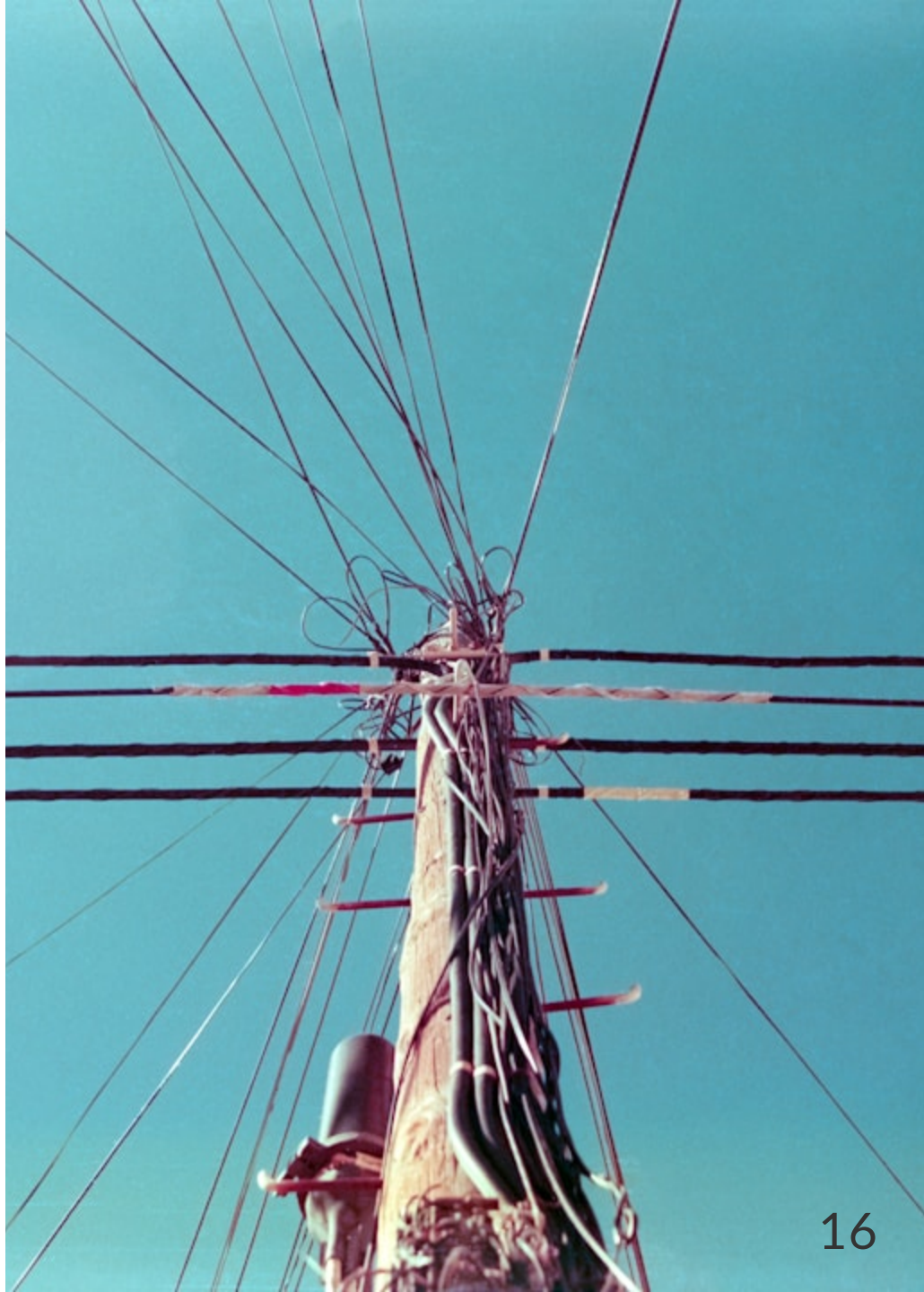
➡️ GitHub Discussions

You can use reactions to express your opinion on a comment!

What will you do next?

In the next chapter, you will learn the following topics:

- Java TCP programming
  - How to send an email with Java
  - How to create a TCP server
  - How to create a TCP client
  - How to handle multiple clients with concurrency



Sources

- Main illustration by Mathew Schwartz on Unsplash
- Illustration by Aline de Nadai on Unsplash
- Illustration by Kelly Sikkema on Unsplash
- Illustration by Carl Nenzen Loven on Unsplash