Odgovori na zadatke 1-3:

1Zadatak SCP

import paramiko

def generate\_ssh\_key\_pair():

# Generate an SSH key pair

key = paramiko.RSAKey.generate(2048)

# Save the private key to a file

private\_key\_path = 'private\_key.pem'

key.write\_private\_key\_file(private\_key\_path)

# Save the public key to a file

public\_key\_path = 'public\_key.pub'

with open(public\_key\_path, 'w') as f:

f.write(f'ssh-rsa {key.get\_base64()}')

print(f"SSH key pair generated. Private key saved to '{private\_key\_path}' and public key saved to '{public\_key\_path}'.")

generate\_ssh\_key\_pair()

Objašnjenje:

Kod koristi biblioteku paramiko, koja pruža SSH funkcionalnost u Pythonu.

Definiramo funkciju generate\_ssh\_key\_pair() za enkapsulaciju procesa generiranja ključa.

Poziv paramiko.RSAKey.generate(2048) generiše novi RSA par ključeva sa dužinom ključa od 2048 bita.

Privatni ključ se pohranjuje u datoteku pod nazivom private\_key.pem pomoću metode write\_private\_key\_file() objekta ključa.

Javni ključ se sprema u datoteku pod nazivom public\_key.pub pisanjem ključa u OpenSSH formatu, sa prefiksom ssh-rsa.

Konačno, funkcija ispisuje poruku koja potvrđuje uspješno generiranje i spremanje para ključeva.

Napomena: U ovom primjeru koristimo biblioteku paramiko radi jednostavnosti. Međutim, postoje i druge biblioteke i alati dostupni za generisanje SSH parova ključeva u različitim programskim jezicima.

C++ verzija:

#include <iostream>

#include <fstream>

#include <libssh/libssh.h>

void generate\_ssh\_key\_pair() {

// Generate an SSH key pair

ssh\_key key;

ssh\_key\_new(&key);

ssh\_key\_generate(key, SSH\_KEYTYPE\_RSA, 2048);

// Save the private key to a file

const char\* private\_key\_path = "private\_key.pem";

ssh\_pki\_export\_privkey\_to\_file(key, nullptr, nullptr, nullptr, private\_key\_path);

// Save the public key to a file

const char\* public\_key\_path = "public\_key.pub";

ssh\_string public\_key\_string = nullptr;

ssh\_pki\_export\_pubkey\_base64(key, &public\_key\_string);

const char\* public\_key\_base64 = ssh\_string\_to\_char(public\_key\_string);

std::ofstream public\_key\_file(public\_key\_path);

public\_key\_file << "ssh-rsa " << public\_key\_base64;

public\_key\_file.close();

std::cout << "SSH key pair generated. Private key saved to '" << private\_key\_path << "' and public key saved to '" << public\_key\_path << "'." << std::endl;

// Clean up

ssh\_key\_free(key);

ssh\_string\_free\_char(public\_key\_base64);

}

int main() {

generate\_ssh\_key\_pair();

return 0;

}

Objašnjenje:

C++ kod koristi biblioteku libssh za generiranje SSH para ključeva i rad sa SSH funkcionalnošću.

Funkcija generate\_ssh\_key\_pair() generiše novi RSA par ključeva dužine 2048 bita.

Privatni ključ se sprema u datoteku pomoću funkcije ssh\_pki\_export\_privkey\_to\_file().

Javni ključ se izvozi u base64 kodiran niz pomoću funkcije ssh\_pki\_export\_pubkey\_base64().

Javni ključ kodiran base64 se upisuje u datoteku.

Na konzolu se ispisuje poruka koja ukazuje na uspješno generiranje i spremanje para ključeva.

Funkcija main() poziva gene\_ssh\_key\_pair() da izvrši proces generiranja i spremanja para ključeva.

Čišćenje memorije se izvodi pomoću odgovarajućih libssh funkcija.

Napomena: Da biste kompajlirali i pokrenuli C++ kod, morate se povezati sa libssh bibliotekom i osigurati da je biblioteka instalirana na vašem sistemu. Specifičan proces kompilacije i povezivanja može varirati u zavisnosti od vašeg razvojnog okruženja i operativnog sistema.

--------------------------------

2Zadatak SCP

import paramiko

import scp

def encrypt\_file\_with\_scp(file\_path, destination\_host, ssh\_public\_key):

# Establish an SSH connection to the destination host

ssh\_client = paramiko.SSHClient()

ssh\_client.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh\_client.connect(destination\_host)

# Create an SCP client

scp\_client = scp.SCPClient(ssh\_client.get\_transport())

# Encrypt the file using SCP and transfer it to the destination host

encrypted\_file\_path = f'{file\_path}.enc'

scp\_client.put(file\_path, encrypted\_file\_path, preserve\_times=True, mode='0600')

# Close the SCP client and SSH connection

scp\_client.close()

ssh\_client.close()

print(f"File '{file\_path}' encrypted and transferred to '{destination\_host}:{encrypted\_file\_path}'.")

# Usage example:

file\_path = 'path/to/source/file.txt'

destination\_host = 'example.com'

ssh\_public\_key = 'path/to/public\_key.pub'

encrypt\_file\_with\_scp(file\_path, destination\_host, ssh\_public\_key)

Objašnjenje:

Kod koristi paramiko i scp biblioteke za SSH i SCP funkcionalnost, respektivno.

Funkcija encrypt\_file\_with\_scp() uzima putanju datoteke, odredišni host i SSH javni ključ kao ulazne parametre.

Uspostavlja SSH vezu sa odredišnim hostom koristeći dati javni ključ.

On kreira SCP klijenta koristeći SSH transport.

Datoteka se šifrira i prenosi na odredišni host koristeći put() metodu SCP klijenta. Šifrovana datoteka je sačuvana pod istim imenom, ali sa ekstenzijom .enc.

Na kraju, funkcija ispisuje poruku koja potvrđuje uspješno šifriranje i prijenos datoteke.

3Zadatak SCP

import paramiko

import scp

def decrypt\_file\_received\_via\_scp(received\_file\_path, private\_key\_path):

# Establish an SSH connection to the remote host

ssh\_client = paramiko.SSHClient()

ssh\_client.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh\_client.connect('example.com')

# Create an SCP client

scp\_client = scp.SCPClient(ssh\_client.get\_transport())

# Download the encrypted file from the remote host

local\_file\_path = 'decrypted\_file.txt'

scp\_client.get(received\_file\_path, local\_file\_path)

# Load the private key

private\_key = paramiko.RSAKey.from\_private\_key\_file(private\_key\_path)

# Decrypt the file using the private key

decrypted\_content = private\_key.decrypt\_file(local\_file\_path)

# Print the decrypted content

print(f"Decrypted content:\n{decrypted\_content}")

# Close the SCP client and SSH connection

scp\_client.close()

ssh\_client.close()

# Usage example:

received\_file\_path = 'path/to/received/encrypted\_file.enc'

private\_key\_path = 'path/to/private\_key.pem'

decrypt\_file\_received\_via\_scp(received\_file\_path, private\_key\_path)

Objašnjenje:

Kod koristi paramiko i scp biblioteke za SSH i SCP funkcionalnost, respektivno.

Funkcija decrypt\_file\_received\_via\_scp() uzima putanju primljene datoteke i putanju privatnog ključa kao ulazne parametre.

Uspostavlja SSH vezu sa udaljenim hostom.

On kreira SCP klijenta koristeći SSH transport.

Šifrirana datoteka se preuzima sa udaljenog hosta pomoću metode get() SCP klijenta i sprema se kao decrypted\_file.txt u lokalnom direktoriju.