# group_13_alina_balint

November 12, 2025

## 1 Data Engineering 1 Assignment 1

Group members are Alina Imanakhunova (2508355) and Bálint Décsi (2506626).

### 1.1 1. Generate keys as `ceu.edu`

Create `key_generation.sh` file with the following content:

```
ssh-keygen -t rsa -f $(pwd)/ceu_key -N ''
```

Then make sure it is executable, and run it.

[1]:
```
!chmod +x key_generation.sh
```

[2]:
```
!./key_generation.sh
```

```
Generating public/private rsa key pair.
Your identification has been saved in /home/balin/repos/ceu/de-1/ceu-cloud-
class/assignment/1/ceu_key
Your public key has been saved in /home/balin/repos/ceu/de-1/ceu-cloud-
class/assignment/1/ceu_key.pub
The key fingerprint is:
SHA256:IErNl2ywnll5uqqaOJ6KvOCVE2PMUwXZRspDLOkDV/k balin@DESKTOP-AG18IMK
The key's randomart image is:
+---[RSA 3072]----+
|    =+*o         |
| . *oBo=         |
|  = *=@ .        |
| .o=.B.E         |
|  .B= . S        |
|  . =  .         |
|.  +  .          |
|B.o ..           |
|OOo..            |
+----[SHA256]-----+
```

`ceu.edu` then publishes the public key `ceu_key.pub` so it is visible for `visitor`.

### 1.2 2. Encrypt a message as `visitor` using the public key

`visitor` downloads the public key and uses it to encrypt a message.

[3]:
```
from Crypto.Cipher import PKCS1_OAEP
from Crypto.PublicKey import RSA

#Import public key
PUBLIC_KEY = "ceu_key.pub"
with open(PUBLIC_KEY, "r", encoding="utf8") as key_file:
    public_key = RSA.import_key(key_file.read())

print(f"Public key:\n{public_key.export_key().decode('utf-8')}")

#Encryption of visitor's message
short_secret_message = "Hello, I am MSBA student in CEU".encode("utf-8")
public_key_cipher = PKCS1_OAEP.new(public_key)

encrypted_message = public_key_cipher.encrypt(short_secret_message)
print("Encrypted message:")
print(encrypted_message.hex())

ENCRYPTED_MESSAGE_FILE = "encrypted_message.bin"
with open(ENCRYPTED_MESSAGE_FILE, "wb") as f:
    f.write(encrypted_message)
```

```
Public key:
-----BEGIN PUBLIC KEY-----
MIIBojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA6gajL5zghrsPUsyzPO2o
TW23MuzbiwOayJ/syqYTm9nvji4/WDOmLw1DSeUqw4ihKTeVUWzxBISWLU2NEWXU
g/cv1Tw0FEFjXoDGM4kaKRbZSDeHtVA9Hptz48RV8EzpuI44ggnfwy5alqZTUh5m
evdRJr8D901zHmxJlDKRsuXT1v7ZLL37qQ0ZvN3wql7pcD835e9mRmXa3GXAmCf7
TO440KNFdJ1oR5+Yafzy3fEvpacZt3OWAjCQkv7sPn9TZWQaj5oP8NC2GNxW78lg
nzysnkZzMx8b6f9s5IdjBR7QkyWqRoip5CBVx9me1nFGTr7LDaAfA6KnVyV2HB9P
/JdLZ1rOKqNKX33MvbXKJLlAYLC7qp13KTKZqXBE4bD4sq011zhXnw+P9a09KC4Y
B9AJQJRi8wcsXqOx5wdrVTe/4vLrq36nklJ03PMRVFNc8jL0VbbfI0prBVyqzG5l
b4hGiUeinpUTA7jBjrZ3U7TCcDcRuuTcMEVpU5c4qhJ3AgMBAAE=
-----END PUBLIC KEY-----
Encrypted message:
07bae715b7edba05aed5f888d3c60109699d7016a0dccaee9e04b6c2519b66808610e13b7ea37c8d
f2ab86c32e2f6c62bf3688e0c2b8598c7a325b26b1da9f276b9c1b9e272d90db382bc2c51541851c
9c7c30031e79b91b183a9f70697cec8bae2d1120bbb4ba596d262d173698a1e21eca232ce42086c1
5c011cf3eb30064039cbf06b1ac6adb77019424d78a850872df1c1fa621d014d06a7eb750e6e5b68
3f0a1b0ac876b7256a7acb92971364cd3dfc676149b039e3ec0c00a5d48c4e5f8ecfa6aca87cefbf
8d97b3027203171cb1538a6cc4b0564e905f53520669a82118e2bc7b3b43455e8556759fab052b36
cf3e177b4b95fae47cd2a218c6ae070a08d088fed3195121a5279cf65c878dce1fb0d00c281b28c0
774568b89acd4609b7178d9355d0b7344795240d38c183967bd13d69a8c0af349c1242b004350baa
2fc8e1199dd2033128728979ab38d46a7890d56e2666cf0d9d2e58e2903d6d92f847470730cfc33f
64a300f8779d22d244901e9484d4174b7e2d2f4faa703975
```

Then sends the encrypted message to `ceu.edu`.

### 1.3 3. ceu.edu decrypts the message using its own private key

[4]:
```
#Import private key
PRIVATE_KEY = "ceu_key"
with open(PRIVATE_KEY, "r", encoding="utf8") as key_file:
    private_key = RSA.import_key(key_file.read())

print(f"Private key:\n{private_key.export_key().decode('utf-8')}")
```

```
Private key:
-----BEGIN RSA PRIVATE KEY-----
MIIG5AIBAAKCAYEA6gajL5zghrsPUsyzPO2oTW23MuzbiwOayJ/syqYTm9nvji4/
WDOmLw1DSeUqw4ihKTeVUWzxBISWLU2NEWXUg/cv1Tw0FEFjXoDGM4kaKRbZSDeH
tVA9Hptz48RV8EzpuI44ggnfwy5alqZTUh5mevdRJr8D901zHmxJlDKRsuXT1v7Z
LL37qQ0ZvN3wql7pcD835e9mRmXa3GXAmCf7TO440KNFdJ1oR5+Yafzy3fEvpacZ
t3OWAjCQkv7sPn9TZWQaj5oP8NC2GNxW78lgnzysnkZzMx8b6f9s5IdjBR7QkyWq
Roip5CBVx9me1nFGTr7LDaAfA6KnVyV2HB9P/JdLZ1rOKqNKX33MvbXKJLlAYLC7
qp13KTKZqXBE4bD4sq011zhXnw+P9a09KC4YB9AJQJRi8wcsXqOx5wdrVTe/4vLr
q36nklJ03PMRVFNc8jL0VbbfI0prBVyqzG5lb4hGiUeinpUTA7jBjrZ3U7TCcDcR
uuTcMEVpU5c4qhJ3AgMBAAECggGAXnxnjmWsUTldKnTzOPpLJVfSy4DN8wZ1i+Ll
27vJ1vbavXD2qknQItcb/83Cwkd7qgSDCOkPNl/BsnDJL+kJMG6wpUs1S6hK0nDz
lQMZyNUszPd+lznaM21YEPHIMMkc5CKntfj+mvMwJ/rnURRtE+CepyIgG8ztWa8m
1erE4JHiQb+LN4FNBif/6D1DWcYQQf2EOBW8GVguUSXT6Jrb32efEL2a4UZcP+6z
vrcd53XlYuBghazpN+Ebvf2mrGUDJREHKT/HLngqZ+9xEa8jxV0KDtOBsxG6e1ax
WxsHOMGiOD2Bztgtuk+P+5cgOYMNQiwNJAKefU5NLMGgjrc5+6EH4GQwHw9sVJIY
b8It8WtiKFFKS3+sWYcLo3IBN/y41j0onnuoNUgp71cWz7LJ854+GdmnYKf0sje0
voFgvne8D02YNUaSnSengRzHSsG0LqQzltWjHTzlkcYDGM1xZlm/7Q/duC5uKO3I
dLR8tsgtDkFHuX5hUv1opoGRK3WBAoHBAPAEMMkcVn2m4iCOFV55tQyCuDrgE0j+
MuNfqdqZZAjEQHSLVPjRFuS0g+H+U8oH+MWNhXyO6JKCTplMbNggkBmAfQ8PXXdS
HTSshBuiEdcRUuNXjRQaJq9oDGXNn0/lkeor9cFDzscPeUn4Wi7sYxRY6T29PPA5
91uhUcjLfqVVV5xxnwWcIUOx7NuO7gpsAvNiJLBdXbbhvOsRmy5I/seOd/b9A7le
kWbvB3wGhDS+FVS7AqJCEVdQOg+EW2lI5wKBwQD5nFJSDaUwJDVCtTAuMt0G22Wg
JMXw++XraBrK9g/Sgsg3X5sJlgQOCYnisRyuB3s6iqLy8RFKazT98ktmhH1feu1y
JUbzVMgLXqURvV0rPSigQtOsl3CVxvECLYjL012fE5/fHWApOlbq2t5/Bt3zHXq5
CsMmJB0wtcJ4z2Dnh791871kTN+DWmaPsCP9YmONCL/ng7889nMPOqTRQ3MUsS8P
1geUBN7O3cKQGslVBM2hZwaYF9UW96BPKLuc5/ECgcAUkydjdgxsBQxJYiSuzpwY
kHQeh+rfTPsrxp2W4aNTpQ8pbnBATBw8SsUrcFPac8h9lt4kiVOii40VUxnZhSEV
LpCJ4/VNLzrVsxw4CjKhTxjd6DltybgbPT4i0/eeL73MZyxP/vIQ2mC5WJLShu0P
Z3fHFRvONAniTvfl0JvwrFowTQUmw/WOCgiV4bDCL/QozPb8L2TDMqKxO/cjhppg
nPPntlTFZTftrPvlQ+G9a7YEZ2uYYTV/WJ9BiUUUx20CgcEA6sC+GzWCGhizjp1h
RtHHVd+ZYFe1YcNGpXMvX5qznsQpEQr1gLp57cnCyFTSKEoY9yHTxrA6Fsdk+0CY
8T3Bu76c+jFc3RTrXuYQUPLLYRMP1kmDUlrLI9IKcIXqleefsTB/txKlkaCwhgh6
IPWQYEllVgptJr3oRhtrXruK47WjqHrnsCEsRRkMRC7gajnSbp2//jjF7svJwpTg
1X6eCOPq4vXkAq3h7qRZi7Xli/QMgLOoW/puCj+xoUEp0ZPBAoHBAJvcCPpEU+/V
ihwxl4ru/VSDCDRbwPhd2c/tEndlJEIk9glfdxXZkBxLWYWJPorTD1Q1PW/waFDL
hj7GK1QiuromGicr42YdO7dS+ohl9hmnBO//k/eKdq0rbNQs2XGDEsLLfxNh+Zyq
TWYVZLffxH4IK5Odko9huaFhOiWYuQvmasUnf/vG+UMdzha+5F12ZpF3V9rndHgy
+j69oIgFwcEDIPEyDVKizqrnY+jKWib2kTUGFqQ2NdWgG3gHF85F+g==
```

```
-----END RSA PRIVATE KEY-----
```

[5]:
```python
ENCRYPTED_MESSAGE_FILE = "encrypted_message.bin"

with open(ENCRYPTED_MESSAGE_FILE, "rb") as f:
    encrypted_message_from_file = f.read()

private_key_cipher = PKCS1_OAEP.new(private_key)

decrypted_message = private_key_cipher.decrypt(encrypted_message_from_file)
print(f"Decrypted message: {decrypted_message.decode('utf-8')}")
```

```
Decrypted message: Hello, I am MSBA student in CEU
```