Kaunas University of Technology
Faculty of Mathematics and Natural Sciences

Cryptology

1st laboratory work report

Variant No. 01

Mairus Arlauskas 2022-11-20
Student MGDMI-

Assoc. prof. dr. Kęstutis Lukšys
Lecturer

Kaunas, 2021

1. Task 1

1. Generate 32 bits long RSA keys. Present all parameters in decimal form and specify which of them are public and which are private. Check by hand whether the generated keys are valid and all necessary mathematical relations holds.

Private-Key: (32 bit)

modulus: 3703726441 (0xdcc26169)

publicExponent: 65537 (0x10001)

privateExponent: 704869941 (0x2a037635)

prime1: 62303 (0xf35f)

prime2: 59447 (0xe837)

exponent1: 47415 (0xb937)

exponent2: 18719 (0x491f)

coefficient: 41339 (0xa17b)

I used python for verification of parameters:

```
import math

modulus = 3703726441
publicExponent = 65537
privateExponent = 704869941
prime1 = 62303
prime2 = 59447
exponent1 = 47415
exponent2 = 18719
coefficient = 41339

fi = (prime1-1) * (prime2-1)

print(f'prime1 * prime2 is equal to modulus: {prime1 * prime2 == modulus}')
print(f'fi is equal to: {fi}')
print(f'fi and publicExponent are relatively prime: {math.gcd(fi , publicExponent) == 1}')
print(f'publicExponent * privateExponent % fi yra lygus vienam: {publicExponent * privateExponent % fi == 1}')
print(f'fi and coefficient are relatively prime: {math.gcd(fi , coefficient) == 1}')
```

```
prime1 * prime2 is equal to modulus: True
fi is equal to: 3703604692
fi and publicExponent are relatively prime: True
publicExponent * privateExponent % fi yra lygus vienam: True
fi and coefficient are relatively prime: True
```

2. Task 2, 3

Generate 2048 bits long RSA keys and certificate request. In the report present ONLY THE PUBLIC part of the keys (remember to rename and securely save the keys' file)

Modulus : 3938848957

publicExponent: 65537