RSA

Neste exercício iremos estudar o funcionamento da criptografia assimétrica com o auxílio do OpenSSL.

a. Criando par de chaves

Para criar um par de chaves pública/privada utilizando o algoritmo RSA com 2048 bits, podemos utilizar o seguinte comando:

```
# Gera um chave privada
openssl genpkey -algorithm RSA -out private_key.pem -pkeyopt rsa_keygen_bits:2048

# Gera um chave pública a partir da chave privada
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

b. Visualizando chave a chave pública

Após gerar o par de chaves, podemos visualizar a chave pública e utiliza o próprio OpenSSL para descobrir o par (n,e). A seguir, podemos visualizar o módulo (n) e o exponente (e):

```
$ cat public_key.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2CvERHyB8Y73LC5YetRB
Z2s7OSh4XTHj5y1+ClgOUtB8MRI3/iA3jiuFem5rsqcboC1jLqeYSdEMAbRmkNov
wW91ocOodzANOTMBh3zYTO2i3mSqAgwLV1UCZq2c3HKg5XftlsgjAIiGe9uYaBY1
PitfYqL65d7X7IowJZkuqx5aylh2txDjKfHoE8Ervc2u7cc2JwVumMAdyysAbWGy
eJGgCR+J4/aSJU0+hI/DD8QKZtj7lCHV+yXmJpkq+SEz9Bh7O7Xch/gfJAcIk6zj
LfmpaB1XUq959v3YkgUW4DxbJ0vU9Or4KaXciso/LOySZ6QjqgYem12+YBxcMYSY
NQIDAQAB
-----END PUBLIC KEY-----

$ openssl rsa -pubin -inform PEM -text -noout < public_key.pem
Public-Key: (2048 bit)
Modulus:
    00:d8:2b:c4:44:7c:81:f1:8e:f7:2c:2e:58:7a:d4:
    41:67:6b:3b:39:28:78:5d:31:e3:e7:2d:7e:0a:58:
    34:52:d0:7c:31:12:37:fe:20:37:8e:2b:85:7a:6e:
    6b:b2:a7:1b:a0:2d:63:2e:a7:98:49:d1:0c:01:b4:
    66:90:da:2f:c1:6f:75:a1:c3:a8:77:30:0d:39:33:
    01:87:7c:d8:4f:4d:a2:de:64:aa:02:0c:0b:57:55:
    02:66:ad:9c:dc:72:a0:e5:77:ed:96:c8:23:00:88:
    86:7b:db:98:68:16:35:3e:2b:5f:62:a2:fa:e5:de:
    d7:ec:8a:30:25:99:2e:ab:1e:5a:ca:58:76:b7:10:
    e3:29:f1:e8:13:c1:2b:bd:cd:ae:ed:c7:36:27:05:
    6e:98:c0:1d:cb:2b:00:6d:61:b2:78:91:a0:09:1f:
```

```
    89:e3:f6:92:25:4d:3e:84:8f:c3:0f:c4:0a:66:d8:
    fb:94:21:d5:fb:25:e6:26:99:2a:f9:21:33:f4:18:
    7b:3b:b5:dc:87:f8:1f:24:07:08:93:ac:e3:2d:f9:
    a9:68:1d:57:52:af:79:f6:fd:d8:92:05:16:e0:3c:
    5b:27:4b:d4:f4:ea:f8:29:a5:dc:8a:ca:3f:2c:ec:
    92:67:a4:23:aa:06:1e:9b:5d:be:60:1c:5c:31:84:
    98:35
Exponent: 65537 (0x10001)
```

c. Criando uma chave 128 bits

Agora, vamos criar uma chave com 128 bits em hexadecimal.

```
openssl rand --hex 16
623a44b53cd64e6a436891dcefb78e23
```

d. Crifrando chave

Podemos obter a chave pública do professor e com ela, cifrar a chave criada no item anterior (c).

```
$ cat chave_publica_miani.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvp9ogCmJ96d51EIDdkPl
nD4rWT+2qcmhrv1l14bLCQD3mG89IhiDMmbN3j0NK7HYxJSPRRR3hNIa7JbC3Bda
S5qFpphwsqw7eYIJojqSJJEnBmrg0p96GAZyF9cAZkN4ZXW5mvi7PAh/CMi6P/34
SOLnqvwgDfT8FVVYJTsYWjXLKbCF3a42VA/F8Ze2ZyrI5V047yZBNbqrNxHF8ppw
vpDEKEvOUu6lM5GKrvIJTlNja7mWjV7deLW5CHKgx6tjyGnF1gRm4+NGMUUaSj4O
XVmipAC5w8kxixpM0y8nl6CEMt2qC8LXh2R1UHPlJWg0K5bYLl1qxittSrtAP4g1
5QIDAQAB
-----END PUBLIC KEY-----

# Crifrando chave gerada no passo anterior
echo -n "623a44b53cd64e6a436891dcefb78e23" | xxd -r -p | openssl rsautl -encrypt -pubin -ink
```