

Segurança Informática e de Redes de Computadores

Nome: Mateus Fernando Jó Matazeus

LAB1

1. Geração de Chaves RSA de Vários Tamanhos:

- 512 bits:

```
C:\Users\Mateus Matazeus>openssl genrsa 512
-----BEGIN PRIVATE KEY-----
MIIBVQIBADANBgkqhkiG9w0BAQEFAASCAT8wggE7AgEAAKEAs39mzFyrXPjPoJlG
dYN+UELzty1+TvcIH956gR4+2fppp53ep8de8M4WFIq0CLQ46t3u+Fd0vP1HOj3W
ISFVWwIDAQABAKBU3pkxawQ05/UlI+Wfr30tyWpIoxcEphxR6zSLFkzrbQJk07KL
Zmn5uQqkzZp8T1G/gts/GXzvw2LhuBv/g4hA1EA7qF3phw4aFitMcpX9dpK/tr4
UQJFT+A/K93xI7+gYYsCIQDAKD1wQFp1RkVzO2qSaZTr6u4Lsu1d+2rxjkyreP21
cQIgPXZtcBNZxF0cr+bSeehaMMc8kJAeALCQpd8b00+e2ECIQCeAPjy7ir7vmtY
LSW6M3YlymQ/JzMmB1lvEXMptCKBkQIhAMRcRaXVez7HK3DrAkKwIA2LwovypSaT
rhv2Ev1TxdGH
-----END PRIVATE KEY-----
```

Executando o comando `openssl genrsa 512` gerou com sucesso uma chave RSA de 512 bits exibida na imagem acima.

- 512 bits:

```
C:\Users\Mateus Matazeus>openssl genrsa 1024
-----BEGIN PRIVATE KEY-----
MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwgGJdAgEAAoGBANp4xBx/rCt8maE2
lPp5ie8b3dS0LY+J7N1657uPry2zV/fCwD31QZ8QHEsMXTgCMmZ4cFvGIi2Mty
QkGjC87nq12Lek+NQru69EVO+Cn7Jr31MGInGzx2YkofmP1hpPX6o/46g58odZG
cvxzXa7fwh2bjCUq1+RnMm9WgdzAgMBAAECGyBbcIX0eSNwmBgdF6FHHGDLVCVjU
JJ0B3LpnI/6qpDnn39aDxnAR1Aj/TAeA414yjpF1JN/T+FLzOC6RCG0e4UVBMDWo
f1iaI9QfttS2Rj51WSMBrrXDVAe2Samuyb8yLLA3PUNM1Jwkf1uJVTgw07j1DyoD
cwsxZJ/w37npQd1CMQ3BAPS3QEGC9/nUajFqUIUJF1f4iyqQDS3V2mRZzoLXGZM
oTz/FKEfxRjYgXpgC6GR1VEvaQmjgnbR9HcaUo+yn50CQDk17d41TP1ESmL4wTB
+bhPW0Ghg08163mfByPdnQZXL5FduetZT8PW3gbCQdnCwQI7dazG7/46sm7d+B9
DF5PAKEAUC0jIysazuBBFA1zxGxuc1BS21nUUMbeY8l0ptpgoFZ6Lp1IFH01E1x
wVR/670mEBAKKcKedR9/bfRXRuc8BQ3JaeTi0w4g89H1j1r9ZgnXOFbwxK0bmYTp
gHegE+a6uzjw15AgryNlg4Dtq7CQ02HxcSQ098kh0va2D0L1bw8jHPw3BANkmb9p1
sP1h6GIACVC1PVnLE8g/NDRWrYysj1p5g1K53y9CQLtE7TDm3puXkf079Vj6Txku
9wG+q4wM/BHDtNo=
-----END PRIVATE KEY-----
```

Executando o comando `openssl genrsa 1024` gerou com sucesso uma chave RSA de 1024 bits exibida na imagem acima.

- 2048 bits:

```
C:\Users\Mateusz.Matazeus>openssl genrsa 2048
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQFAASCBKwggSIAGeAaoIBAQQDPuUIAIEzJegBS
5i0cBcb+qEBR1t8hd73ATz57yBSu+ts0xNHNRRPpWuEGlGf5NcHf-dgTeSv5SvN
8F5u5u5dZ370AT5u2039V0Tvm10B4N10EA3bzuzks8Bmm1qZ
fHlyJ7mZMoKVVnnRUMkmy0z03hmJ1Fe2A6BuejruTUVGfFqgT2uT205HDBPwyl6m+
0FSyZtY+IFKlGqyYdgLtpN0ko2qCnGx6bqbpFScEQaAYAvnYz+FiYCnPOV0+6tSh
501b24tAnrurAgE11omhEEfygzAe83SL20ppa2k3T5HPY8CVSPK9DVXr/OEGzPqM
w1RlN5yJagMBAAECggEAD69g/Jd0/h7uLQZ0pDnmXmdwnk/4PW01EGHxbG6gPK
0z0lPlKKg03ATanCepdncgtB0xmAjCtDxtCRKZVX14qFXeX4LElGjngaoj/UUjX
YlGb0QT091QB5pp7d1JzCtFoupK09okKLenCvBVYH8Kf36THfK1VK5DLtaW7Av
ZvZgmZ5R8BKXk7WmPRbRD70DfFmdw0nBdhPqHqE1o35YvB44+6FQ8T1vvK5sm
4u+qPierEgFby7Y+SkW6E1LvgD0YfHudqYfDxGL1MVAFeqghB0k8F1FMf6
10m8B09Qd0eqgrZwFtHbPhazr/McXZHX/qPcaqb+QK8q0Dxw01anB68c1L7D5
gDgt4lFokQ0KwI7ILU7yZjZdrL9C9LSk1Qn8k1InCFetGey3GxC8qvz2a1/L5k
Qzb2osWzULf3CDVGRzKRxvTg4f6t80J3RnQu048HuN/Sax17vhOpKc5nx9mgatJh
qu0eGN91E18cxunV/KULQZtzwKBQd0Gmcfnaaf1Mb0zzfrrwI/kNITxm+VHfQ0NC1
5m/R+35fHzH23sXY1RDALMnnRLXMhVki1f3fE2/qxzKwVIA/rR8qe+VHQ6kmcrt
zHawfCuW5317Sg8shAeasuWhR3r2eR2N5Y5EE1/HT1bmm4MgaQe/xSv96Kwuy
Q0A0u3v7QK8bQD08nf+/B2hRu0S/3VvfyGIMnAeEc+EOl0n6tyLqs9GS28AFG
MI7FnuaqxRfGnu0JyTWm7xcxf+RfAVJ3YTrFsc0fYygnDwS25dsSk42z1/cF
KcPv2A40000000000000000000000000000000000000000000000000000000
Rv4ARhZzLWskAJ2RP2006BQw0A0NDs0dazPqWk0kT0Ughn8qurscWk1e0X0U
1C1YZ1+ADW1JvF5DK0F9TL0utezajm729J0w+1P1InUanR500BUs90LYFrrnAik
k+1qD0akFTLcFLeA4PTKRor1afK5bMo6vgrgNdih+QK8bQC2DvHvdD7FUOEh8K
2u1a9MAPySYUMHThn9M25x2397oy+v1mQmrd1qdEQ719AFCagVOT23aj1AggdRA
/AMLvCeUxY99fmV35NweJ+fk713z10710Bc1o6T4H17Gup2xDNJCSZx/h198jdW1
WlRyHfW8Xo0ZxAPALqu8c+WAA==
-----END PRIVATE KEY-----
```

Executando o comando `openssl genrsa 2048` gerou com sucesso uma chave RSA de 2048 bits exibida na imagem acima.

Para fazer a compracao com a algoritmo Diffie–Hellman (DH) fez-se a execucao do comando `openssl genhd 1024` e obteve-se o seguinte resultado:

```
C:\Users\Mateusz\Documents>openssl gendh 1024
Invalid command 'gendh'; type "help" for a list.
```

Segundo pesquisas feitas, conclui-se que o erro acima ocorre pelo facto do comando `openssl gendh` não está disponível nas versões mais recentes do OpenSSL. Recomenda-se fazer o uso do comando `openssl dhparam` para gerar chaves Diffie-Hellman (DH). Fazendo o uso do comando `openssl dhparam 1024 -out xxx` obteve-se a chave:

```
-----BEGIN DH PARAMETERS-----
MIIGAgEBANst6xeEHdyD3ymjlhp7e8UvLWljeJaWCCsMQSf2FYBueSQ50xaM
fjKXhz2vjKMW64R19Gw0VIEomDK0SE089OGXjbAyShnpqX+e7FV3baFL21+J4
2/nm1nI3dnUK8ADEtoPC7ANLopXmXxFW/rBm3s9OX9y6hfW0PvnAgECAGIARu==
-----END DH PARAMETERS-----
```

O comando `openssl dhparam 1024 -out xxx` gerou a chave Diffie-Hellman de 1024 bits acima e o salvou em um arquivo de nome "xxx". Pode-se conferir a chave gerado na imagem abaixo.

```
File Edit Format View Help
-----BEGIN DH PARAMETERS-----
MIGfGA0GARNvMjE2MDUyMDUyMDUyMDUyMDUyMDUyMDUyMDUyMDUyMDUyMDUyMDUy
WGU9QUYRAYc2PHK159ECfmaCfNz29HphNaL33Jf0/dOpLmZu8GT7/4n1sX9wFh
fiSLtB+gVur4Jop+XpR0moxXuCwXUFUsx8RADmmQv8SyL1Iyf2vAgECgAIArw=
-----END DH PARAMETERS-----
```

2.1. Olhando o código BaseRSAExample fornecido na turma, nota-se que o mesmo mostra a habilidade de cifrar e decifrar informações usando um conjunto de chaves geradas usando o algoritmo assimétrico RSA. Ele demonstra como é possível codificar com a chave pública e decodificar com a chave privada, e vice-versa. O código BaseRSAExample ajuda a compreender o funcionamento básico da criptografia RSA.

- As chaves usadas possuem um tamanho de 128 bits, uma vez que o valor do módulo usado que está em formato hexadecimal é `d46f473a2d746537de2056ae3092c451`.
- No código fornecido, as chaves estão sendo inicializadas manualmente nas variáveis `pubKeySpec` e `privKeySpec`. O valor do módulo (N) é `d46f473a2d746537de2056ae3092c451` e o expoente público é 11 (em hexadecimal) e o expoente privado é `57791d5430d593164082036ad8b29fb1`.
- Sim, o tamanho do ciphertext produzido faz sentido. O aumento do tamanho é devido ao uso das técnicas criptográficas, pode-se notar na figura abaixo o resultado

```
C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.1\lib\idea_rt.jar=51599:C:\Program Files\Java\jdk-17\bin" -jar C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.1\bin\idea.exe
input : 1234567878
cipher: 60463babea996119ced952314cdea2e1
plain : 1234567878
cipher: 05396208929eb2401f6ebfd5353468fd
plain : 1234567878
```

- d) Após fazer as alterações referidas, tes-se o resultado abaixo:

```
C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.1\lib\idea_rt.jar=51641:C:\Program Files\Java\jdk-17\bin\java.exe" -Dfile.encoding=UTF-8
input : 1234567878
cipher: 604e3babea996119ced952314cdea2e1
plain : 000000000000000000000000000000001234567878
cipher: 05396208929eb2401f6ebfd5353468fd
plain : 1234567878
```

- e) Se o bloco plaintext for maior que o valor N subjacente à operação mod e às chaves usadas, a operação de cifragem falhará. A cifragem RSA requer que o plaintext seja menor que o valor N para funcionar corretamente. Após fazer os testes, obteve-se:

```
C:\Program Files\Java\jdk-17\bin\java.exe" --javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.1\lib\idea_rt.jar=51655:C:\Progra
Exception in thread "main" java.lang.IllegalArgumentException: Create breakpoint : RSA publicExponent is even
    at org.bouncycastle.crypto.params.RSAKeyParameters.<init>(Unknown Source)
    at org.bouncycastle.crypto.params.RSAKeyParameters.<init>(Unknown Source)
    at org.bouncycastle.jcajce.provider.asymmetric.rsa.BCRSAPublicKey.<init>(Unknown Source)
    at org.bouncycastle.jcajce.provider.asymmetric.rsa.KeyFactorySpi.engineGeneratePublic(Unknown Source)
    at java.base/java.security.KeyFactory.generatePublic(KeyFactory.java:345)
    at BaseRSAExample.main(BaseRSAExample.java:22)
```

BaseRSA > [BaseRSAExample] main 39:35 CRLF UTF-8 4 spaces

- f) No código fornecido na turma, não está sendo usado padding. Isso pode levar a problemas de segurança. O padding ajuda a garantir a segurança e a integridade da mensagem criptografada, prevenindo ataques como o padding oracle, onde um atacante pode inferir informações sobre a mensagem original ao analisar as respostas da decifragem.

2.2. a) O código RandomKeyRSA gera um par de chaves RSA aleatórias de 8192 bits usando KeyPairGenerator. O código anterior utilizava chaves com valores fixos para fins de demonstração, enquanto este código gera chaves aleatórias.

b) O desempenho da geração de chaves do código RandomKeyRSA e do tempo de computação (cifra/decifra) aumenta significativamente devido ao similar aumento do tamanho da chave. Ou seja, quanto maior for a chave, maior será o tempo de geração de chaves. Na figura abaixo verifica-se o grande tamanho do ciphertext:

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.1\lib\idea_rt.jar=51666:C:\Progra
input : 000056785678
cipher: 2ea2a983baf4c08e550aba5ee18902443c2999e34c7bbafe05375e1fd063a3f93e03cdf026b36201395adf77d5bf8456067419c7901374a99cca4e681ddffdc120483c884f3783ce829
plain : 56785678
cipher: 71862c7224a8ae94660a6d15486c42dab117939c7198d2474d1132cc20bb0a8a7a3ad7fad7b0e3162db3590f219dd4fe96edbbe75e700a38a1ce35140c956ce0de699570d20a076222f
plain : 56785678
```

2.3. a) O código PKCS1PaddedRSA utiliza a forma V1 do padding PKCS1. A escolha de usar o padding PKCS1 é importante para garantir que as mensagens tenham uma estrutura padronizada antes da criptografia, o que ajuda a evitar vulnerabilidades e ataques conhecidos em RSA.

2.4. a) O tamanho do ciphertext obtido é o esperado. Isso ocorre porque o tamanho do ciphertext em RSA depende principalmente do tamanho da chave RSA e não do método de preenchimento. Pode-se verificar o resultado abaixo:

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.1\lib\idea_rt.jar=51714:C:\Progra
input : 00abcd
cipher: 3c362702bd3c42348b5e7afe8472c3eadee963bc6245d9f9974ad93eb34f44eebedfc25c9a49d590c392c34fde5aea29063ad0913dd2b6b1f0a09742c4aebc04Size: 64
plain : 00abcd

Process finished with exit code 0
```

b)) O programa OAEPPaddingRSA gera um novo par de chaves RSA em cada execução. Isso faz sentido porque a segurança do RSA depende da escolha adequada das chaves. As operações de padding OAEP são realizadas com base nas chaves geradas, garantindo que cada corrida do programa use chaves diferentes para criptografar e descriptografar. Portanto, a inicialização do cálculo de padding faz sentido mesmo ao gerar novos pares de chaves a cada execução.

Exercicio Opcional

- a) No protocolo descrito, a distribuição de KS1 e KS2 é feita com base em envelopes de chave pública. Ou seja, para enviar essas chaves simétricas para o destinatário, o remetente as criptografa usando a chave pública do destinatário. O destinatário, que possui a chave privada correspondente, pode então descriptografar as chaves simétricas e usá-las para comunicação segura.
- b) A e B podem ter razoável confiança de que estão se comunicando com os interlocutores corretos, desde que o sistema de chave pública seja seguro e que a autenticidade das chaves públicas seja verificada.