PRACTICA 2

Cifrado Asimétrico

Matilde Cabrera González

Tareas por realizar:

*** Generad, cada uno de vosotros, una clave RSA (que contiene el par de claves) de 901 bits. Para referirnos a ella supondré que se llama <nombre>RSAkey.pem . Esta clave no es necesario que esté protegida por contraseña.**

Vamos a darle el nombre “matiRSApkeypem”:

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
 openssl genrsa -out matiRSAkey.pem 901
Generating RSA private key, 901 bit long modulus
...++++++
.........................................++++++
e is 65537 (0x010001)
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
```

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
cat matiRSAkey.pem
-----BEGIN RSA PRIVATE KEY-----
MIICFQIBAAJxFz/GhaCVji/UAMW+YQ6DBn/EE0xlPCsxqRSq4vcNQa0YIWf4Xnt3
aFXfe9TsLnewfaPwufOAq9bE8Mwf0qtgiEiligvBJL82LdqE7IDJ4VKcAlNH1bBE
aVRhuW8sX5+rD6xcUutzawE5u+zHko1FLC0CAwEAAQJxDNfiZ3dqpv2s9DBJF3zY
zQJP4wlU84Q5Dtqn/HHWFO32Yp35CtjSAgVOTC43uwPLr9yc41ytPECG7kjKVcN8
Awh1btr5tpoYIU0ufkR3B4I5TeiU4YjQDxF/Vom+WE+7QoV4k/IFv+++3+o/SMWQ
VwECOQeeDEGDf7Nso4wSTVJ+fke5kkRkZcKVBoh0cwtzCP1JgEQFVtzd13rLoF/J
34UdzQqlSz5ZMLnzsQI5Aw1XlI3rmZo2lwapL9GzESnFwBjfDpE3gg87/483oQMj
LnnLLGRLkk+jXqSiMVI6VHuDP9VmNYs9AjkB2hXGs66PM4FLY4Oy45fV3jLGN9ZZ
iRuOfykdBnmQApeRce7dQX3Kl8jEoPKbjVQ1G1yAqvRisPECOQEFjFWsFDu8xZ1U
m3noZlrOcuXU7YYA0mX1F456sJJzbGl4V6yqfimiAvwBfFGqRuBU4CCUHLy7uQI5
BYdqVa3oTQItYjPwUeG+SRoRyBNUlpVWlAuUGvcwlTQKRzt60ZQCbB2YnRaJKlPN
uahUcXVlsB/D
-----END RSA PRIVATE KEY-----
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
```

*** “Extraed" la clave privada contenida en el archivo <nombre>RSAkey.pem a otro archivo que tenga por nombre <nombre>RSApriv.pem . Este archivo deberá estar protegido por contraseña cifrándolo con AES-128 . Mostrad sus valores.**

Extraemos la clave privada y protegemos el archivo anterior que contiene la clave privada con contraseña “0123456789”:

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
 openssl rsa -aes128 -in matiRSAkey.pem -out matiRSApriv.pem
writing RSA key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
```

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
cat matiRSApriv.pem
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,90CC9F88303691308F24CFC1A30D7681

ESrveCYT93uEDVsU5FiSFDz4FXyF5A/lLr/FRF3+TT61heS/NrzOpWBhfhZ+nQoT
OznT9SXst808k/JYTNLJ58FuZBQLLUIKprJkUpoDGGjnSgiO+cWZ47/RsiqaoCep
TmkscLtd9bvyp/8zE48CTZBjy/ABgWSqKt9vWo9xyLeGu58dLcuCs+ikI8ejUY+H
qLUbXVVPC8fyHor/lBldLfyFlzKOXXWa7VUIMvMAKz8EVPNECBTAYPX1hr5iBhUu
eQxrDpzVMeaMruobhqBgeXJaED5wElFskXk2lIicSz9aAl0KI+vWaekWU1b9fbSg
70AlboxtK4YGBQwdaReZpo5I1V2P+mZrqvSytFslQHpnfCNWBprNoGLSBSnKKE6s
0IcyS4w7CM57/f0ERxh2F36rJh9Y3UyJn2fT/Xw4Gomazh0R9moLg9Zy9JlbosVf
Lgs5opLctLSMMSxDTjTnf/XxKE1VP7AzkMstnJG+ekbJdJctONLVwb+y91H/icnd
B784LJ9sPF4UvfWrQa7rfOt00UV3I51ik7kzZVEqiBskgtefAue6W7nU5SKhzfU8
k7lHNmVnhsLJX+9FHv6hu9LQCdtYWDbOUp42uhXW3u2E7piWol6qfXpAUO3ZuPUY
R47jnzKEV1pdZZpB0WrvTkA9cTP8FFXgD3y9skirR4oGqGm/uQdhPptFzO1SToHP
C9xcbA/Wz0wPV6VjfgiODQ==
-----END RSA PRIVATE KEY-----
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
```

Muestro sus valores:

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
 openssl rsa -in matiRSApriv.pem -text -noout
Enter pass phrase for matiRSApriv.pem:
Private-Key: (901 bit)
modulus:
    17:3f:c6:85:a0:95:8e:2f:d4:00:c5:be:61:0e:83:
    06:7f:c4:13:4c:65:3c:2b:31:a9:14:aa:e2:f7:0d:
    41:ad:18:21:67:f8:5e:7b:77:68:55:df:7b:d4:ec:
    2e:77:b0:7d:a3:f0:b9:f3:80:ab:d6:c4:f0:cc:1f:
    d2:ab:60:88:48:a5:8a:0b:c1:24:bf:36:2d:da:84:
    ec:80:c9:e1:52:9c:02:53:47:d5:b0:44:69:54:61:
    b9:6f:2c:5f:9f:ab:0f:ac:5c:52:eb:73:6b:01:39:
    bb:ec:c7:92:8d:45:2c:2d
publicExponent: 65537 (0x10001)
privateExponent:
    0c:d7:e2:67:77:6a:a6:fd:ac:f4:30:49:17:7c:d8:
    cd:02:4f:e3:09:54:f3:84:39:0e:da:a7:fc:71:d6:
    14:ed:f6:62:9d:f9:0a:d8:d2:02:05:4e:4c:2e:37:
    bb:03:cb:af:dc:9c:e3:5c:ad:3c:40:86:ee:48:ca:
    55:c3:7c:03:08:75:6e:da:f9:b6:9a:18:21:4d:2e:
    7e:44:77:07:82:39:4d:e8:94:e1:88:d0:0f:11:7f:
    56:89:be:58:4f:bb:42:85:78:93:f2:05:bf:ef:be:
    df:ea:3f:48:c5:90:57:01
prime1:
    07:9e:0c:41:83:7f:b3:6c:a3:8c:12:4d:52:7e:7e:
    47:b9:92:44:64:65:c2:95:06:88:74:73:0b:73:08:
    fd:49:80:44:05:56:dc:dd:d7:7a:cb:a0:5f:c9:df:
    85:1d:cd:0a:a5:4b:3e:59:30:b9:f3:b1
prime2:
    03:0d:57:94:8d:eb:99:9a:36:97:06:a9:2f:d1:b3:
    11:29:c5:c0:18:df:0e:91:37:82:0f:3b:ff:8f:37:
    a1:03:23:2e:79:cb:2c:64:4b:92:4f:a3:5e:a4:a2:
    31:52:3a:54:7b:83:3f:d5:66:35:8b:3d
exponent1:
    01:da:15:c6:b3:ae:8f:33:81:4b:63:83:b2:e3:97:
    d5:de:32:c6:37:d6:59:89:1b:8e:7f:29:1d:06:79:
    90:02:97:91:71:ee:dd:41:7d:ca:97:c8:c4:a0:f2:
    9b:8d:54:35:1b:5c:80:aa:f4:62:b0:f1
exponent2:
    01:05:8c:55:ac:14:3b:bc:c5:9d:54:9b:79:e8:66:
    5a:ce:72:e5:d4:ed:86:00:d2:65:f5:17:8e:7a:b0:
    92:73:6c:69:78:57:ac:aa:7e:29:a2:02:fc:01:7c:
    51:aa:46:e0:54:e0:20:94:1c:bc:bb:b9
coefficient:
    05:87:6a:55:ad:e8:4d:02:2d:62:33:f0:51:e1:be:
    49:1a:11:c8:13:54:96:95:56:94:0b:94:1a:f7:30:
    95:34:0a:47:3b:7a:d1:94:02:6c:1d:98:9d:16:89:
```

*** Extraed en <nombre>RSApub.pem la clave pública contenida en el archivo <nombre>RSAkey.pem . Evidentemente <nombre>RSApub.pem no debe estar cifrado ni protegido. Mostrad sus valores.**

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
 openssl rsa -in matiRSAkey.pem -pubout -out matiRSApub.pem
writing RSA key
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
 openssl rsa -in matiRSApub.pem -text -noout -pubin
Public-Key: (901 bit)
Modulus:
    17:3f:c6:85:a0:95:8e:2f:d4:00:c5:be:61:0e:83:
    06:7f:c4:13:4c:65:3c:2b:31:a9:14:aa:e2:f7:0d:
    41:ad:18:21:67:f8:5e:7b:77:68:55:df:7b:d4:ec:
    2e:77:b0:7d:a3:f0:b9:f3:80:ab:d6:c4:f0:cc:1f:
    d2:ab:60:88:48:a5:8a:0b:c1:24:bf:36:2d:da:84:
    ec:80:c9:e1:52:9c:02:53:47:d5:b0:44:69:54:61:
    b9:6f:2c:5f:9f:ab:0f:ac:5c:52:eb:73:6b:01:39:
    bb:ec:c7:92:8d:45:2c:2d
Exponent: 65537 (0x10001)
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
```

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
cat matiRSApub.pem
-----BEGIN PUBLIC KEY-----
MIGMMA0GCSqGSIb3DQEBAQUAA3sAMHgCcRc/xoWglY4v1ADFvmEOgwZ/xBNMZTwr
MakUquL3DUGtGCFn+F57d2hV33vU7C53sH2j8LnzgKvWxPDMH9KrYIhIpYoLwSS/
Ni3ahOyAyeFSnAJTR9WwRGlUYblvLF+fqw+sXFLrc2sBObvsx5KNRSwtAgMBAAE=
-----END PUBLIC KEY-----
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
```

*** Reutilizaremos el archivo binario input.bin de 1024 bits, todos ellos con valor 0 , de la práctica anterior.**
*** Intentad cifrar input.bin con vuestras claves pública. Explicad el mensaje de error obtenido.**

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$ openssl
rsautl -encrypt -in input.bin -out input_cifrado -inkey matiRSApub.pem -pubin
RSA operation error
139652600406464:error:0406D06E:rsa routines:RSA_padding_add_PKCS1_type_2:data too la
rge for key size:../crypto/rsa/rsa_pk1.c:125:
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
```

El error nos dice que los datos son demasiado grandes para el tamaño de la clave. Esto sucede porque en RSA los datos a cifrar se asignan a un entero, para que funcione este entero debe ser mas pequeño que el bloque RSA usado. Si estamos usando un bloque de tamaño 901 bits, los datos deberían

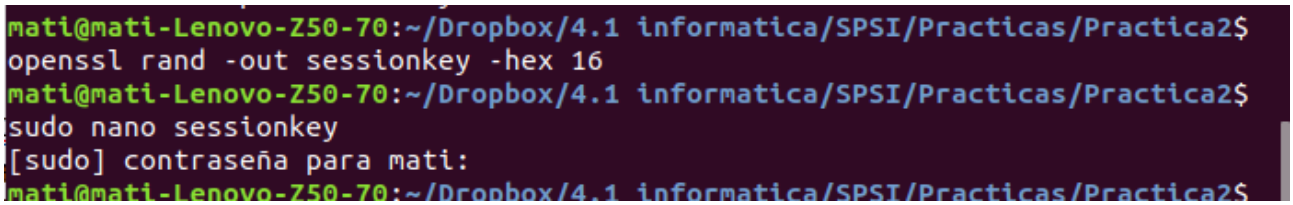
dividirse en trozos de 901 bits como máximo y cifrar cada fragmento de forma secuencial (usando modos ECB o CBC). Esto es muy lento de hacer, las soluciones que hemos encontrado es primero hacer un cifrado simétrico y luego cifrar con la clave RSA la clave utilizada para el cifrado simétrico. También hemos encontrado una solución con la orden smime, de la siguiente forma:
openssl smime -encrypt -aes128 -en input.bin -binary -outform DEM -out LargeFile_encrypted publickey.pem

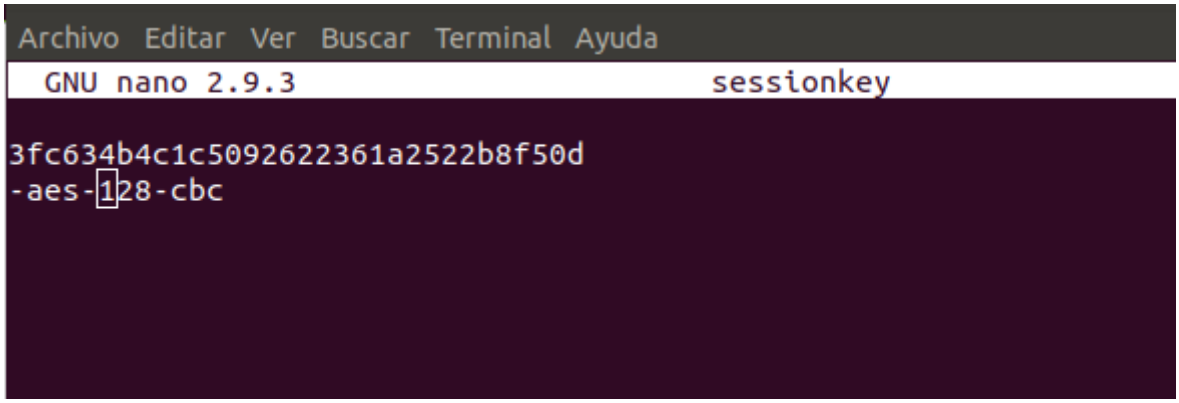
*** Diseñad un cifrado híbrido, con RSA como criptosistema asimétrico. El modo de proceder será el siguiente:**

1. El emisor debe seleccionar un sistema simétrico con su correspondiente modo de operación.

Vamos a usar -aes128-cbc

2. El emisor generará un archivo de texto, llamado por ejemplo sessionkey con dos líneas. La primera línea contendrá una cadena aleatoria hexadecimal cuya longitud sea la requerida para la clave del criptosistema simétrico. OpenSSL permite generar cadenas aleatorias con el comando openssl rand . La segunda línea contendrá la información del criptosistema simétrico seleccionado. Por ejemplo, si hemos decidido emplear el algoritmo Blow sh en modo ECB, la segunda línea deberá contener -bf-ecb .





3. El archivo sessionkey se cifrará con la clave pública del receptor.

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$ openssl
rsautl -encrypt -in sessionkey -out sessionkey_cifrado -inkey matiRSApub.pem -pubins
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
```

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
xxd sessionkey_cifrado
00000000: 15fa e3cf 76c6 9610 7004 6f88 70df 7494  ....v...p.o.p.t.
00000010: 7795 7872 564a a420 d1c0 4c53 e337 6746  w.xrVJ. ..LS.7gF
00000020: 9822 9858 e3cc 5954 bced 8279 500f 9e89  ."X..YT...yP...
00000030: dfad d0f6 1fc6 861e 9bc7 9a23 811f 7200  ...........#..r.
00000040: 6945 648c 8bb3 6152 f0e0 f5fb 579e 0195  iEd...aR....W...
00000050: af8c cbe8 58e1 ba65 8fdc 01c2 43d1 8ab9  ....X..e....C...
00000060: 4ef7 5627 e05a f6f8 d7dd af4c 92f9 0079  N.V'.Z.....L...y
00000070: 2d                                       -
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
```

4. El mensaje se cifrará utilizando el criptosistema simétrico, la clave se generará a partir del archivo anterior mediante la opción -pass file:sessionkey .

Entiendo por mensaje el archivo que anteriormente intentábamos cifrar y nos dio un error, “input.bin”

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$ openssl enc
-aes-128-cbc -in input.bin -out output -pass file:sessionkey
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$ xxd output
00000000: 5361 6c74 6564 5f5f 6025 be1d 6bef 2288  Salted__`%..k.".
00000010: 2bfe 9abc cfeb f389 0a0d 6c78 57bd 082d  +.........lxW..-
00000020: 3772 e0a8 b908 2145 6c04 6113 2979 f590  7r....!El.a.)y..
00000030: da3e f4a7 8d41 75de 44b5 823c d541 2566  .>...Au.D..<.A%f
00000040: ba2c 2a93 a59c 7394 f2ce b453 cbc5 b188  .,*...s....S....
00000050: c749 51ea dd34 5fef f131 daf9 def6 5374  .IQ..4_..1....St
00000060: 6f09 a39c 5560 86ab 226e 0b44 f0b8 53ec  o...U`.."n.D..S.
00000070: 9ffb d9e3 0144 e706 8cd5 65c9 5b01 9a1c  .....D....e.[...
00000080: f98f a7c4 29a3 6ebe c6ff 4bbf 28e6 14d7  ....).n...K.(...
00000090: 3fe6 88ce 1504 18f8 ee20 ba3c 1f9a f994  ?........ .<....
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
```

*** Utilizando el criptosistema híbrido diseñado, cada uno debe cifrar el archivo input.bin con su clave pública para, a continuación, descifrarlo con la clave privada. comparad el resultado con el archivo original.**

Hacemos la misma operación del ejercicio anterior pero para descifrar “session_cifrado” .

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$ openssl rsautl
-decrypt -in sessionkey_cifrado -out sessionkey_descifrado_privado -inkey matiRSApriv.pem
Enter pass phrase for matiRSApriv.pem:
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$ xxd sessionkey_
descifrado_privado
00000000: 3366 6336 3334 6234 6331 6335 3039 3236  3fc634b4c1c50926
00000010: 3232 3336 3161 3235 3232 6238 6635 3064  22361a2522b8f50d
00000020: 0a2d 6165 7331 3238 2d63 6263 0a         .-aes128-cbc.
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$ cat sessionkey_
descifrado_privado
3fc634b4c1c5092622361a2522b8f50d
-aes128-cbc
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
```

El archivo obtenido lo usamos para descifrar output, tenemos que añadir la opcion -d para descifrarlo, sin el volvería a cifrar y no obtenemos la salida esperada.

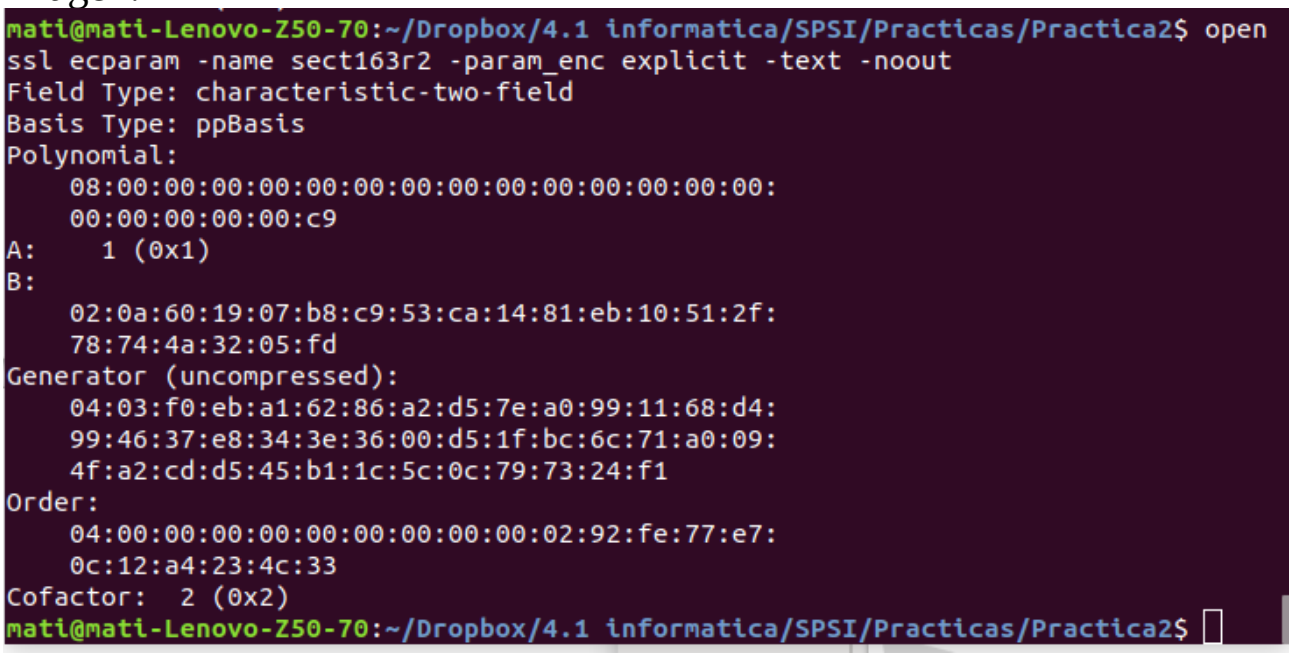
```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$ openssl enc -aes-128-
cbc -d -in output -out input_descifrado.bin -pass file:sessionkey_descifrado_privado
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$ xxd input_descifrado.
bin
00000000: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000030: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000040: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
```

*** Generad un archivo stdECparam.pem que contenga los parámetros públicos de una de las curvas elípticas contenidas en las transparencias de teoría. Si no lográis localizarlas haced el resto de la práctica con una curva cualquiera a vuestra elección de las disponibles en OpenSSL . Mostrad los valores.**

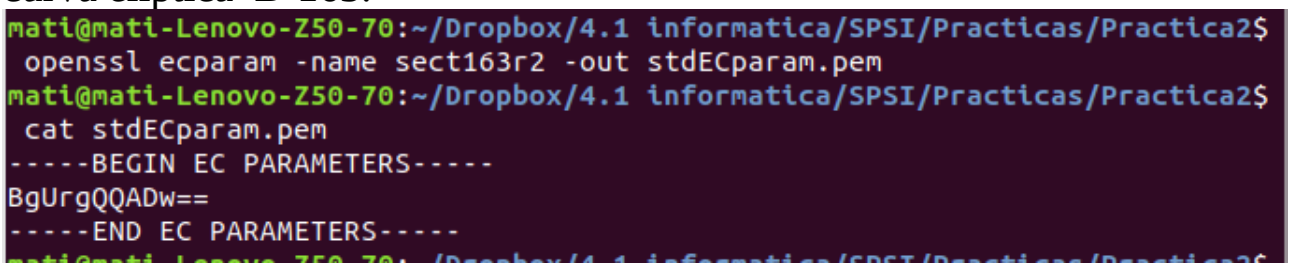
Muestro las curvas con “openssl ecparam list_curves”, como la salida es demasiado extensa y me cuesta encontrar las curvas vistas en clase de teoría, P-192 y B-163, asi que busco con la orden grep las salidas que contienen esos números.

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$ openssl ecparam -list
_curves | grep 163
  sect163k1 : NIST/SECG/WTLS curve over a 163 bit binary field
  sect163r1 : SECG curve over a 163 bit binary field
  sect163r2 : NIST/SECG curve over a 163 bit binary field
  c2pnb163v1: X9.62 curve over a 163 bit binary field
  c2pnb163v2: X9.62 curve over a 163 bit binary field
  c2pnb163v3: X9.62 curve over a 163 bit binary field
  wap-wsg-idm-ecid-wtls3: NIST/SECG/WTLS curve over a 163 bit binary field
  wap-wsg-idm-ecid-wtls5: X9.62 curve over a 163 bit binary field
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$ openssl ecparam -list
_curves | grep 192
  secp192k1 : SECG curve over a 192 bit prime field
  prime192v1: NIST/X9.62/SECG curve over a 192 bit prime field
  prime192v2: X9.62 curve over a 192 bit prime field
  prime192v3: X9.62 curve over a 192 bit prime field
  brainpoolP192r1: RFC 5639 curve over a 192 bit prime field
  brainpoolP192t1: RFC 5639 curve over a 192 bit prime field
```

Hemos comprobado cada una de las curvas mostradas hasta dar con la curva B-163 de las transparencias de teoría, lo mostramos en la siguiente imagen:



Generamos el archivo stdECparam.pen que contiene los parámetros de la curva eliptica B-163.



*** Generad cada uno de vosotros una clave para los parámetros anteriores. La clave se almacenará en <nombre>ECkey.pem y no es necesario protegerla por contraseña.**

*** “Extraed" la clave privada contenida en el archivo <nombre>ECkey.pem a otro archivo que tenga por nombre <nombre>ECpriv.pem . Este archivo deberá estar protegido por contraseña. Mostrad sus valores.**

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
 openssl ec -in matiECkey.pem -aes128 -out matiECpriv.pem
read EC key
writing EC key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
 cat matiECpriv.pem
-----BEGIN EC PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,339CBF20CC31A4842D70EF6DCED17BCE

5UqH3SccRAyLiBxjoCwBLnuI8knUvOsoSKb12P7eLg/lt1fetXLtUP0RxkZnGHak
NdZdDCdY+ejNr5U5JTWz86nC1QdkeYJl4sMerzibgZys8wayI78u7r790ZK8I8zf
-----END EC PRIVATE KEY-----
```

Para que esté protegido por contraseña usamos -aes128 y le ponemos la contraseña indicada en el pdf de practicas “0123456789”

*** Extraed en <nombre>ECpub.pem la clave pública contenida en el archivo <nombre>ECkey.pem . Como antes <nombre>ECpub.pem no debe estar cifrado ni protegido. Mostrad sus valores.**

```
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
 openssl ec -in matiECkey.pem -pubout -out matiECpub.pem
read EC key
writing EC key
mati@mati-Lenovo-Z50-70:~/Dropbox/4.1 informatica/SPSI/Practicas/Practica2$
 cat matiECpub.pem
-----BEGIN PUBLIC KEY-----
MEAwEAYHKoZIzj0CAQYFK4EEAA8DLAAEApYMcQUptTVOl4sQdYGRzb4RROKzAOzp
0t+0FPvXb7e32NoeCBoE2RPx
-----END PUBLIC KEY-----
```