SEGURIDAD Y PROTECCIÓN DE SISTEMAS INFORMÁTICOS

Grado en Ingeniería Informática

Universidad de Granada

Practica 2 Criptosistemas Asimétricos

4 de noviembre de 2018

Índice

En este documento se explica la segunda practica de la asignatura, que trata de aprender a utilizar la herramienta OpenSSL, y para completarla es necesrio entregar los 11 puntos de la misma, hay que entregar este pdf con los comandos y las capturas de pantalla necesarias para demostrar que se ha realizado la practica.
En esta ocasión trabajaremos con cifrados asimétricos, intentando entender su filosofía de trabajo.

This document explains the first practice of the subject, which tries to learn how to use the OpenSSL tool, and to complete it, it is necessary to deliver the 11 points of the same, this pdf must be delivered with the commands and screenshots necessary to demonstrate that the practice has been carried out.
This time we will work with asymmetric ciphers, trying to understand their work philosophy.

Tareas a realizar

1. 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.

Para realizar esta tarea ejecutaremos en la consola el siguientes comando:

```
[usuario@portatil ~/] openssl genrsa -out emisor/CarlosRSAkey.pem 901
Generating RSA private key, 901 bit long modulus
...++++++
.....++++++
e is 65537 (0x010001)
```

2. '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.

Para realizar esta tarea ejecutaremos en la consola el siguientes comando:

```
[usuario@portatil:~]$ openssl rsa -text -in emisor/CarlosRSAkey.pem | tail -n 14 > emisor
    /CarlosRSApriv.pem
writing RSA key
[usuario@portatil:~]$ cat emisor/CarlosRSApriv.pem
-----BEGIN RSA PRIVATE KEY-----
MIICFQIBAAJxGMck+/D0wOTbi2TEYwPlnqfq/w1rzBFURQDxm1UcRddkkaULSYMa
dBJBuOZBZQTmW/IIsrnWFhk4zbzfFxutWbW6bQOleEjhnbIjpEenb2osvLlDmbW3
+hCb1Mp4AsT8hMOUk8AdXA5aXWJTxWsYCe0CAwEAAQJxBGCaPhAxjSphRt0EfOGh
XUgZEfoTAzhEhPRHusNsfUIyODDKL7CDqeHXzoyPHQOilrfJpZmDIuBL7rcWcSqn
cXoNy/KcTLNGlVRqk2pg16aAqPNdrUWQs5+j8xKc6LF7I/jfRGg/OU1/Qu1gP7sm
ZAECOQdQl+J5uYJ7M/gH9nIzEIuARx46f1c1pAD3qh5p4Ablb4gQqoLW6ioD6spp
2qDAKYbJddozMkILgQI5A2Mp6f3lOjYTwGebgCSjUyN/oW5y6laIYuHY07dY2qrB
MRHEELFOuzXtfqBr7d72XrwE0Knk2yRtAjkAkqMSLQ4F1bgAGD+IOc913Rm0mEgY
gC7TlJFRyGCuqZDOSt/PVx0HJT9hkmB4bhPSZjxAVfn9igECOQHIiD2HjNIbvERU
h5psL4xKqRwNMPTMf9tHOcckxOPCeBlkWAvNvTx0hMdgn9HIy35OkF5mw3TX9QI5
Ajr/n6oQ3LI4Q2SwKVZSlJ31tmyOyE2jEm+UkyuV88NfzRsEhEKp7MFyyEUbvYwB
RpnHzXYi2gVm
-----END RSA PRIVATE KEY-----
[usuario@portatil:~]$ openssl enc -aes-128-ecb -pass pass:mipass -nosalt -in emisor/
    CarlosRSApriv.pem -out emisor/CarlosRSApriv.pem.enc
[usuario@portatil:~]$ hexdump emisor/CarlosRSApriv.pem.enc
0000000 7513 42bc 477f 94a4 b16a 8738 2ffb d156
0000010 8fb9 1235 3847 db83 b632 534a 7cd7 c9f0
0000020 c727 d4fd 9298 487c 0cc5 6765 3661 d90c
0000030 4f77 4698 5fb1 5c9e 6d28 be81 0c6c 320f
0000040 ba90 4ca2 b32f 1881 de8f a52d bd65 8a1f
0000050 80d4 54f6 e982 03b2 01c4 0fa2 865f 906a
0000060 c64e 9eb1 190d 88a4 6a6d 58ab f87f fb4d
0000070 25d4 ca5a b61a 4c8f 86f8 9da3 0bef 78f7
0000080 8c3a fbe6 c919 de68 de5b b3ed 3263 920a
0000090 915a 9250 df9e 1ce0 03e2 f67f 6874 0dd9
00000a0 c362 5937 4627 1a26 fdf4 75cc f673 67c6
00000b0 33a6 a7d2 fe52 b3a6 43d7 2d27 e1a6 03ea
00000c0 ea31 e006 3369 9b0b 17d3 7f11 eb72 7655
00000d0 4d9d c1dd 4dcc e33b a233 8b67 b6a5 3481
00000e0 6b95 1374 9944 4525 44c7 4e7f bf15 c2e1
00000f0 9c6d 5e8c f791 d57e dd4a 688e 0d47 468a
0000100 7b2e 75bc aa3c adc0 25bb 64c1 3d99 61b3
0000110 c920 6c41 297d 1b0d c7e0 1531 76b7 42c7
0000120 12bd 8c92 64c6 645d eb29 d3db 1bfc a36b
0000130 1fe4 5be6 9689 ced7 449d b1ed 0c12 d0b7
0000140 0da7 5f0d b118 d78c 3c6f a55e 3c17 e196
```

```
0000150 493b f5bc ad73 fecb 1b5a ddb0 2544 0d67
0000160 4786 b5d5 8c4e 8e8b 2160 95dd 25bc 156f
0000170 d655 6631 7915 6d21 9fcb 7243 960a 92df
0000180 9f9b 4e2f 4dfe 6943 d84a aeaa f365 b066
0000190 523b 7bcc 709a c3dc 07ee 585e 0c50 fe54
00001a0 9d46 edf7 2ac7 3d64 fb38 0e7a 4e71 5ce8
00001b0 8cc8 8735 1df1 103e 454b a4fe 7040 9492
00001c0 868d 16a9 3ab9 0028 68f9 d195 b422 3d66
00001d0 85e7 c7fd 6ba9 674c c5c0 c4c1 d11f a335
00001e0 8f62 190b c0d6 8bd2 3aab 556e 60cb bb28
00001f0 40de e003 ebf1 020e f2b0 75a3 dcf6 32f0
0000200 38f5 4167 b97d 8c41 5f01 7d6d fda2 c1bc
0000210 0f9d cf2e c394 8cd4 cffe f88e 0394 6ec1
0000220 efe3 17b8 165a 5a70 3dc7 1b66 1c4d 28a1
0000230 e403 6159 8a3e c78f c775 3b17 3579 449a
0000240 431e ad9f a089 1c3c 9a4c 0891 ccde 4f13
0000250 6174 9cbf 8859 fe63 d632 332c f8fa 31af
0000260 9d99 89eb 1d8f b2a9 4b1a 0069 9e0d 9903
0000270 8d68 26a3 fcb2 d938 ef46 48a3 673d 2a29
0000280 1b26 5f4a 8f82 76a1 f9ff b86b 6172 0340
0000290 b632 1fd7 8019 9603 2328 67b4 7d4b 7d0e
00002a0 6c6c cd33 d82c 13ff c008 f7d9 27af 9488
00002b0 be72 a33f 1cea c014 2c3a 7e6d a8f7 bb3d
00002c0 4e96 f815 603e 41b1 433c f64f 97f4 5138
00002d0 9efc 1c0c 027c 8c7f 0eea a082 b05f c856
00002e0 4f0c e85f 7d59 c9f4 ad03 4f97 f4b8 4f18
00002f0 26f5 fd13 da07 0a53 1042 3540 7609 c76c
0000300 1a31 aa90 1553 5410 c1cc 366c 61f8 686b
0000310 776a 5fe7 cdcd e400 0983 c6b7 e3ec 3ea7
0000320
```

3. 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.

A continuación se muestran todos los comandos utilizados para realizar este apartado:

```
[usuario@portatil:~]$ openssl rsa -in emisor/CarlosRSAkey.pem -outform PEM -pubout -out
    emisor/CarlosRSApub.pem
writing RSA key
[usuario@portatil:~]$ cat emisor/CarlosRSApub.pem
-----BEGIN PUBLIC KEY-----
MIGMMA0GCSqGSIb3DQEBAQUAA3sAMHgCcRrTMxWTEYoR4Wx9qS42cCL3anDzEKGG
F1RhepUFey2iePAt68Vfrc7e/b+6w3iEXQ3jZjNAxphrFQshNM8h7DCeYeVR9ijt
IRMthMz9mnvUrscbSVcX5ybq+/10fCw6S/sR8DOqojCwia6NGKf/kPdbAgMBAAE=
-----END PUBLIC KEY-----
```

4. Reutilizaremos el archivo binario input.bin de 1024 bits, todos ellos con valor 0, de la práctica anterior.

A continuación se muestran todos los comandos utilizados para realizar este apartado:

```
[usuario@portatil:~]$ dd if=/dev/zero of=/tmp/imput.bin bs=128 count=1
1+0 registros leídos
1+0 registros escritos
128 bytes copied, 0,000317665 s, 403 kB/s
```

5. Intentad cifrar input.bin con vuestras claves pública. Explicad el mensaje de error obtenido.

A continuación se muestran todos los comandos utilizados para realizar este apartado:

```
[usuario@portatil:~]$ openssl rsautl -encrypt -inkey CarlosRSApub.pem -pubin -in /tmp/
    imput.bin -out /tmp/imput.bin.enc_rsa
RSA operation error
139730000234304:error:0406D06E:rsa routines:RSA_padding_add_PKCS1_type_2:data too large
    for key size:crypto/rsa/rsa_pk1.c:125:
```

Esta claro que el problema es que el fichero que estamos intentando cifrar, contiene mas bits, que nuestra clave y por lo tanto no puede cifrarlo, de ahí que se utilicen algoritmos hibridos para cifrar información.

6. Diseñad un cifrado híbrido, con RSA como criptosistema asimétrico.

Partiendo de las siguientes restricciones:

1. El emisor elegirá AES como criptosistema simétrico
2. El emisor creará el fichero *sessionkey* que contendrá en su primera linea una linea de 128 valores hexadecimales.
3. En la segunda linea del fichero anterior el emisor seleccionara el metodo simetrico por el cual se cifrará el mensaje.
4. Se tendrá que cifrar este fichero con la clave publica RSA del receptor.
5. Se cifrara el mensaje a enviar con el método simetrico elegido y con la contraseña mediante la opción -file y el fichero anterior.

A continuación se realizara una simulación de como se tendría que utilizar el criptosistema anteriormente descrito.

1. Primero tenemos que generar, el para de llaves para el cifrado asimétrico de la contraseña del cifrado simétrico, par mayor facilidad separaremos la parte pública y la privada en dos ficheros diferentes.

```
[usuario@portatil:~]$ openssl genrsa -out emisor/RSAkey.pem 4096
Generating RSA private key, 4096 bit long modulus

  ..................................................................................++++

............++++
e is 65537 (0x010001)
[usuario@portatil:~]$ openssl rsa -in emisor/RSAkey.pem -outform PEM -pubout -out
  emisor/RSApub.pem
writing RSA key
[usuario@portatil:~]$ cat emisor/RSAkey.pem
-----BEGIN RSA PRIVATE KEY-----
MIIJKwIBAAKCAgEA1DIcCsMWUuJSmntTJPuN/N42lt+KYAMW1KAB7WjZUy4P3U1I
JNHtbf8P/ZOR1fBlPA+lAvTVEECNVmLNBXjOGL0GxDilPsE1lxfBQoeBybb7ulBx
A90xxksBihSe7IYWH8zdsZdBO7jlAd8HVvPzF8ZJlFkkCbAo867aJY29RH3t7HDw
AK912nH6BNH/SYrnCLhg6wfur4bCrSUsCaZCqo9XADrL2NjVD/owdx4BshGZniDP
zkyoqWBrnEO2CW3UzWYkAIm7wsSvkq54uiZoqvpnGzQ4rTKrdqzHAV9WTKHqrt/g
+JAtlKKQXgGIWBKN8T1s/+Cp5hxdWmdWo8z/LrKqmuw7OpALdX5wGtNVPYzI30y3
ZDMnTQGx3BFXaR4M9iKKWPSDO0RMIUSrq+Sq/T7Jcs3Qb8F8/H0p5oae7Pt4f5KR
mC30+o3Saq/cwOLcMn3bkZDtBDWugS1xH31qwsQ7HguG8gbrYBCn6G4QAaiQcIY1
VANa+USRTZqCjJwAgF5o4WuUIsOuoAjdPswWu3+YOTV5z0CHxoIzePx+rAZQL3KL
Xy+hGncRKABQqJrB2RlvSG2GfjZRsFUz+u5m4w2LJhEwYxG05lDoczQd75HzrVRi
TOiuTClLct7at4sKD+dmXQVRLm6Bls6b6pvhITi/jWfI5wPidqPjNPrk2zsCAwEA
AQKCAgEAuGBmD0/eKNywq7x2zU6q3Ep1InlQMgbdIpl5s0EEfQssC72n7x5dcNNM
oKhF8p/0RtDNL5lvPLTqWmJ1BMkoQ0SUCZGyu6mEGCoy6cz2kEAn0Icu/A2Jv2PT
DZjHUntb4/vNsg7cyswAaLTTGtHTqxfHPViLQMPJgZuvvyu+hjHptyWW1PGgtNOO
S/vwF4XTI7eZtv7ofanZpDsFXckSakMsdG2iTGR0FXLuA8KV9JToZwbZqcbgUnQO
bPREy0KcwnB63eDPvffLGufwzMas8cusJW4lxRCe/fo43M9rHCEg2xx6zV/xyxHJ
4XTxdJ2CqlgWSut6lRCWaQQbuy5XlzXr+5wq/Ctz018XqtoM34GR6kvNDcYPNkJA
a4pwe4ggGu/pNzaVTeXhMGu8dsyyqDqcEUGWygj3yS1pzV+kxnyud+06IvDguxW6
```

```
iq1XsAwghUXl6f6IYUQVQIxi0Xru/SeVIygCWV3DtqagoQtpYtcNYbPdcTojm4bS
sZ2wTbOTQhAdo0H0X3c3I6QMQTQoX8bDdOqVZ8vlQ26WXRDOBlS16W26nUWGHx2w
PxksJT+JZoHyjXC0D04mcDS9M0HxBucJv/f9+Wd3FKRduFJr4HbpeiqapyABjWmH
XuZhTNK4v8Qb0U/G/X7d7BjC9VMUiQU8mDc3Yt3fMTbvwDnJ9+kCggEBA0mmJN2w
hL9ABBmdlLCaaWcidebZ9vihXpe4j0caVZf+t7b4XYhObFvxWxELnrc25vWjtHhi
izP333U8UELwnNMSp3Lbj9qN0m5BLyn0Dlb08yIiBqqDtSHiD5VbDqtDT8crmoPX
0NuBGuhQRItFrMo8Oc8SQ6Blb9/AhRE+f+te3wCY6EV+gX8PjpnofERH8qVTMS5y
c3U5wwVBnayYBDcFOJJGUMru4uSLLOPRp7n3Wz+iCb0zafJUjDcsC+eQGW9lZ9HC
mR94cky8lAWJRvpiOJ/C/RzY4H9LnIzXprVHRiaqsiMkSuhRQUOeu/WjXn73tAaU
XBOyQRK84275uHcCggEBAOh+mCQ5zCk9kEOc0WSIWr8vaiiwDgn66LPXh7tjviVH
/lu6KI8e+ueoSjgCRvVVULMts4TYAyaV93TFMyO/jGxd0kfv5hClMPNWTYTOS1Ro
Vt9qw0SXf0txQLlzXN+e5oR6riW25pU2odh7WrvezHdu9qS6GwnW+1mX+I02mgEO
pkGF7NLh2R9wuC6e2l03+gP4L4MHSz6f7iu3HVw2xTdnkISkFDk1cuIaLHdcAOk2
d0WvoX7jl/UuxmgMdMz2OcuaibWRbrKgTudEi3zERYiA+YtvS7hkTR7uFQ/E13sA
eDVb4hKonnbEYwR+sdD7McDz6/AxaLCaTweABt5H6F0CggEBANRHjiEYmlMEf3Ui
dZBKik8YwRP/nmJggofIN+y64V9aHmIPnzMTRWH3k+yVqRr+t7IfjPz/U0fhAdYK
+UUIRQOEZDx3fWhtB+4CLS6pvUljdwTg8zuCtFPXhp8E/vvdCSOE2zrEHs7/qom9
O7lpkUagl0kqZ+TXpGAfEwcKyL/5DxUE16sj99WQNdhhoF2NHMjWH836UCoGU8h3
yCCicNnoVXCYkBkAF4bbZ+qIdiIAuZfDyaNrjeuMo2QREbPOeT17XWr3qZUtInPT
tYyeH62nBZkpWDwYU0d0VASmfaEf432IApbpB2GxSWF9W3oeduFWfeqoaku+3V5/
3QbbvZsCggEBANmYShp1gbADb3/MfemCeLRA0zGbfLg4ffx7twxaEMdhtxzwgS9a
Rmf0KOp8rx7EJ5BbMI4fgetmJkDCo9dMWke+E6XCUGe5w7OrTUgnQC2SrePST0lu
JPRd3b7zyrN8S7EoBL9FjE+Q3y72XHhC0MjjMLIBgHbYH0y+ICFCPibJKC9hAcJp
IEX9wFPZ1vOyBSwb55PRisnk8ws1CaSZmkD9i9OooUjdzlTxPDbGBBBBKqYs95AD
wN/3VDhjmBmQRY1aaJJnt9w0j0SB8a7nH1DfrS6kDpFYRpmxKEMm7qmvSdJe2Zqu
+AieSzIUxaWaw/kcEfEuvRW0hky1tTfqwv0CggEBAMLAf5m5EtxsMIsneA5Z2o0W
jBxfY1haUCo0nkA9k4qFz9p3O8bcPdlQ1DJwtpzKsDfvW6TBGOi5jy2Kqd1mINgI
Dtx5KSeGQklQNWZ9E8cnK2JbJEHo6c18pDH9mwVbXT40D0Facawgj9KOVx0yyzcm
431vSpcGh424ZGACpkFsu8aA7hBnFCJ6UOvbttWoctJw2G1Y36/il4qHrb8/Civo
kxZ6TAyitidK3lmHHFV6+52CYePVI2OwYpPAn6jvs24NCdnx2eul87SdAa28MrzM
45053L5+tNghPRF2dfYMD7bjHj47jNDlK6dQbtvTHjHgWLf7NCLunguewwAxu/o=
-----END RSA PRIVATE KEY-----
[usuario@portatil:~]$ cat emisor/RSApub.pem
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEA1DIcCsMWUuJSmntTJPuN
/N42lt+KYAMW1KAB7WjZUy4P3U1IJNHtbf8P/ZOR1fBlPA+lAvTVEECNVmLNBXjO
GL0GxDilPsE1lxfBQoeBybb7ulBxA90xxksBihSe7IYWH8zdsZdB07jlAd8HVvPz
F8ZJlFkkCbAo867aJY29RH3t7HDwAK912nH6BNH/SYrnCLhg6wfur4bCrSUsCaZC
qo9XADrL2NjVD/owdx4BshGZniDPzkyoqWBrnEO2CW3UzWYkAIm7wsSvkq54uiZo
qvpnGzQ4rTKrdqzHAV9WTKHqrt/g+JAtlKKQXgGIWBKN8T1s/+Cp5hxdWmdWo8z/
LrKqmuw7OpALdX5wGtNVPYzI30y3ZDMnTQGx3BFXaR4M9iKKWPSDO0RMIUSrq+Sq
/T7Jcs3Qb8F8/H0p5oae7Pt4f5KRmC30+o3Saq/cwOLcMn3bkZDtBDWugS1xH31q
wsQ7HguG8gbrYBCn6G4QAaiQcIY1VANa+USRTZqCjJwAgF5o4WuUIsOuoAjdPswW
u3+YOTV5z0CHxoIzePx+rAZQL3KLXy+hGncRKABQqJrB2RlvSG2GfjZRsFUz+u5m
4w2LJhEwYxG05lDoczQd75HzrVRiTOiuTClLct7at4sKD+dmXQVRLm6Bls6b6pvh
ITi/jWfI5wPidqPjNPrk2zsCAwEAAQ==
-----END PUBLIC KEY-----

```

2. Ahora el emisor generará el fichero sessionkey que contiene los datos del cifrado simétrico y una cadena aleatoria de tamaño 128 este fichero será la contraseña de el cifrado asimétrico que utilizaremos para cifrar el mensaje en cuestión.

```
[usuario@portatil:~]$ openssl rand -hex 128 > emisor/sessionkey
[usuario@portatil:~]$ cat emisor/sessionkey
402e49cf3b4789c0c1c700d300b7ed9b97e2bc8c760775b21e9ad4af035bd536837d5083df74d9c30
bb20f8aaf43fb79974c0ea2c82a99f43bb9101a47ff59100fc5fa002d1698433338f2c4d6250d35c5
1a68464dd95bb1a70023862bd96f979bfd2d7a167d7161150bf5c948afa5d79e253ac23dad24179e7
6d100141f021c
[usuario@portatil:~]$ echo "-aes-128-ecb" >> emisor/sessionkey
[usuario@portatil:~]$ cat emisor/sessionkey
402e49cf3b4789c0c1c700d300b7ed9b97e2bc8c760775b21e9ad4af035bd536837d5083df74d9c30
bb20f8aaf43fb79974c0ea2c82a99f43bb9101a47ff59100fc5fa002d1698433338f2c4d6250d35c5
1a68464dd95bb1a70023862bd96f979bfd2d7a167d7161150bf5c948afa5d79e253ac23dad24179e7
6d100141f021c
-aes-128-ecb

```

3. En este punto tendremos que utilizar la parte publica del receptor para poder cifrar el fichero

sessionkey y poder enviarlo al receptor con seguridad de que no se puede ver únicamente por el, para ello vamos a crear los dos ficheros de llaves igual que hicimos antes con el emisor.

```
[usuario@portatil:~]$ openssl genrsa -out receptor/RSAkey.pem 4096
Generating RSA private key, 4096 bit long modulus
......++++
.....................................................++++
e is 65537 (0x010001)
[usuario@portatil:~]$ cat receptor/RSAkey.pem
-----BEGIN RSA PRIVATE KEY-----
MIIJJwIBAAKCAgEAoofBojHA7w8lsftrbqUubSXrlCOJnK3va0P9Sah/gJC6vU5a
TTUBha9sGrAepHclYcwod0KrgQXGTHE+CJsr7nsqfIXmrGpxK6hupl833TLqgYE8
dM4IOPL19OjUbEKoZYAcRR/7jQu5OBqeQNhhtul51Ptra/KsrhIkOTV4K6ELEiKM
N7BgnchMtNyNdnTDj4YS357EN9gSQHHxoEt3fMqp7aocns5+Blj3FvbfUdpeCDRI
Foh5urvd7kHqWpjoGWJ81b8Tz9735i6JTo7k8CjvLQNBwqsAReEFjnEBkXL+/gdC
WqTfBbQIaCS4vkwXCL4+nEChWTquB+qcXqs/5LXWBsdlG3VtCG1rpEhX9hBCGFl7
sqaG/QMxn9tAFyop53vdCY9NkEO1TmRmC7J/HsSpMfG3lTVRC4ZTjRpjOtRu9Ur1
ncPlHNd6t9edV6PeZp41FWkqQYK3BmanJprd7d3pRolpE4j56fueTteEI2bvpiZQ
gVAqKeZvf1CJOFtyuC4TvSX7VSEHjDgKm1OnwyeRl2EMzPsNOMFFt1CMipR4YJSX
oTUxFUHtP7Kbw6hXVhhZUaKFPzZigBGiDpZiTvXOM2wxaoYZOb7rH6ADfU8/aiHE
JG60sCGMtXTYb3YjJV/PSZuOP51YFIVWfTrH2S/PDzIN7Vr3v4OUvyW4sakCAwEA
AQKCAgBfqF8Cwx22lcet1TgJyxZkDylz6uaXXJfJHh75Nz6zwHrn6DGlt0jKEdEi
c/3U9nr98FbwukCOk4N+gjfsgMd+eOcsmvKFD7DD7UISkYAjpE/5IMMgTBLy0ayJ
FG/Rt/m8o4X/+S7oeLnul1wE9lWsfgXCoKc8xXt0iCDhuuKbaxFE/yhfrmNu+fCL
kjOAQIDI/DQngR4jb/xuCaIYodIVlIIm6zQlE4uA2OO0lGAjoM4vLfYvrOtYSDH6
hLkV0WDpb9nhQ7C5a5Wj2EfgAKIl5zG94GjX1iU2+5V1nHU7Od7aTBN1ngvVI3y/
Lr284rO75ZycAj30cmDzL9YQseHMHi2cuA/Gx3ttPRrJOa+S6ZNSx7VAj6wgE5AI
4tCny3QA94GlM39lE4Izy144ZgxjidXrOjIMbB9aGK4j/T1ONZeRuLjMFL+ybWgL
2sl6OwsINpfFQCM8htGLwj6vAruh5lPeprKNUj8inAu5aIg5+3eNuDjzpA8Sq2Jb
vHagAXM/bYYIkrwXh7hcAeoQ37joq74kol0/sBbTEEpHFonl0iTpT1RuMbGquJCQ
dbgS1kA6hG7zdMCS9gT1K5fW6DsiOhSbukcgclfh0IYc/5mMWR2iDpJMJmzW53e0
pLYkjOyS6BXjsg5ZfdXOWlzc2fbuUUXd/MivVbf6y8eVBubEAQKCAQEA0qYPNIwm
Oxas+KMgNV2CShs7+8Uwi4UkwPHDtWAo/Scsb4Zv7rjmn5tvyYk3nmGavcbJ1s2x
8UhCJ/Qp2EQWOi1LtzKqonYIq1Bc6tL6Uu1IWHAmqlVYUbG/iCTfrR2mW6xsyvh4
vj1Cw7tja6F9LrMl2Ivnt/CIE6w0qH3FDh0fqYv0C6lMmtgIGisSMcbDjiByChbo
mx0urfHsnEFG4xCeAKv05tHgKz1Icp9z08JHWW7u7WTXBseaPESzFREggSjgVoBX
cEKG35Ti2fcS01qEeGhS5XLSbjMWMn+TsUHZApan3smq6vZdbjsA7vgfbCEVgaLL
85zdbXdjYAe7YQKCAQEAxYWk3cMQT78ptoz0LqLC01uVELKamloruhygy3gqF+gS
7KOPuwt8kflGBkm6l0hTpGHvXXynJbEVqqtWHwshL2qhUaQRwaas3i2jg6cMm4hd
oZkacZMRbJtH9YJ0Xntbbt9lXjkKQrrQ0vaAkKSkwvb2+UZ49fJ+GapDT34HA2dg
fy9C7OSP1t1upqj2JvAMZB2LYsnPJ3r+9TaX6bpkFLqXumc+JTlLX7mYifh8Ore5
T4m51teP1MJaZqdh7eJGcoTUCNml4ndxv4Gvm5u4mKijosyMeww0RVGhaCHzB/Vy
/JvQzpFfjAZc9aLHMfSSch+oYd9bE3f53/EEtKYjSQKCAQBsPBsP+vsh3VLfJqJE
CVoKzdeSVA3PgogSA6d1qIUnHlu8qi0VUjhdxjY4LaNAZBJodJ39Gw/VVDV55gEi
OEnbdq6y4PvAIjnuHIrBtcodlgJYPrf/SX7p6xTo1VmAz/1MmVoFKIqR5VRK0hwR
j5aMykwl6i0T3ClvQkqEUKtRxzEfouOwojrUTTBMkdj1bpXIj29JM+As41+6zNWw
q/HKdh1ZZwGdIDW4wfJZwwh2xBp5H4DwxaGewCzFzt7dckFrXB95EtTav7QhaZTu
Pf+UTaxHWgNOFiN9sA41wEuZBqaEp/h1htWliRUUsIiYQTGUE6PDSmzVLcKwnmmX
TCHBAoIBACcG6fVbNlzbY42LOjAAvsfuH3zmHUAEMwGZ6Hdrbb29gAe9hg8qDNn2
y8wwQyrjAovHQflRyODcpXBMXD7T/yGUSv2LkmqV2TApClbX6djNT9q/God4TIgO
AmDS/tUwQHeesjO0x3851Erqo6pwrZFMJK5KK1pFKH8/sf8B4rbcdSeMGggefQEy
TUumtMQqx/f8Em9ZepHNbpTazZdD2cRyvcn53kCSmn72d9pC+6DKblX1239Epkme
gSgpSGMtQ0gbUJ3TC0C84v+4yw7O7OFU6Mgg7Fzfo3BS2bZuhst6xtZ8nvxBowRS
uYlbOUp4jXr5IFb+P8uP+BIOV/6tmOkCggEAPm2ej903XxeAHcy41OvJjBkviONa
L9bOivp6jKe7B2BgY2+FcrfdV1x3SrKJNMpGi2kA13eK/A3g+xkCtcEF7jeQ+hDH
3KUeBtq7mueJ2ANeUlT7jJ7RYimopAslfBhkZbb/0jiTsekGOg3ZvtRTHxssjfxp
GfTVoLEx37EL5J8hCjwHE2IWBm59V35+fDFye4Lgok4aASslYPG8xFjwCtpM5qQR
mCqrstKNFA1qbiGJIADYaa4UaBqJwaHnc6mWf+8G0t2Iwmo8ULQgWaQb4G76GT8Z
nEupwEMj4CQWCtY099XWvU0RRSFYrLsaIiGc3xZtmd7rfYifNgWzEyhfEg==
-----END RSA PRIVATE KEY-----
[usuario@portatil:~]$ openssl rsa -in receptor/RSAkey.pem -outform PEM -pubout -
  out receptor/RSApub.pem
writing RSA key
[usuario@portatil:~]$ cat receptor/RSApub.pem
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAoofBojHA7w8lsftrbqUu
bSXrlCOJnK3va0P9Sah/gJC6vU5aTTUBha9sGrAepHclYcwod0KrgQXGTHE+CJsr
7nsqfIXmrGpxK6hupl833TLqgYE8dM4IOPL19OjUbEKoZYAcRR/7jQu5OBqeQNhh
tul51Ptra/KsrhIkOTV4K6ELEiKMN7BgnchMtNyNdnTDj4YS357EN9gSQHHxoEt3
fMqp7aocns5+Blj3FvbfUdpeCDRIFoh5urvd7kHqWpjoGWJ81b8Tz9735i6JTo7k
8CjvLQNBwqsAReEFjnEBkXL+/gdCWqTfBbQIaCS4vkwXCL4+nEChWTquB+qcXqs/
5LXWBsdlG3VtCGlrpEhX9hBCGFl7sqaG/QMxn9tAFyop53vdCY9NkEO1TmRmC7J/
HsSpMfG3lTVRC4ZTjRpjOtRu9Ur1ncPlHNd6t9edV6PeZp41FWkqQYK3BmanJprd
7d3pRolpE4j56fueTteEI2bvpiZQgVAqKeZvf1CJOFtyuC4TvSX7VSEHjDgKm1On
wyeRl2EMzPsNOMFFt1CMipR4YJSXoTUxFUHtP7Kbw6hXVhhZUaKFPzZigBGiDpZi
```

```
TvX0M2wxaoYZOb7rH6ADfU8/aiHEJG60sCGMtXTYb3YjJV/PSZuOP51YFIVWfTrH
2S/PDzIN7Vr3v4OUvyW4sakCAwEAAQ==
-----END PUBLIC KEY-----

```

4. En este punto el emisor dispone de la parte pública del cifrado asimétrico del receptor, por lo tanto se procederá a cifrar el fichero sessionkey que será la contraseña del cifrado simétrico del mensaje que queremos enviarle al receptor

```
[usuario@portatil:~]$ openssl rsautl -encrypt -inkey receptor/RSApub.pem -pubin -
  in emisor/sessionkey -out emisor/sessionkey_enc

```

5. Como el emisor es el que elige cual será el tipo de cifrado que se usará para la parte simétrica del envío, ahora el emisor procederá a cifrar el mensaje que desea enviar al receptor mediante el cifrado simétrico que ha indicado en el fichero sessionkey.

```
[usuario@portatil:~]$ openssl enc -e -aes-128-ecb -pass file:sessionkey -nosalt -
  in /tmp/imput.bin -out /tmp/imput.bin.enc.aes-128

```

6. En este punto el emisor está en posesión de los dos ficheros cifrados necesarios para poder realizar el envío al receptor de forma segura, sin que el mensaje se vea comprometido, hacemos la simulación de envío.

```
[usuario@portatil:~]$ cp emisor/sessionkey_enc receptor/sessionkey_enc
[usuario@portatil:~]$ cp /tmp/imput.bin.enc.aes-128 receptor/imput.bin.enc.aes-128

```

7. Una vez que el receptor tiene los dos ficheros, este puede comenzar el proceso de descifrar el mensaje procediendo primero a descifrar el fichero que le indicará cual es el método simétrico utilizado para cifrar el mensaje, y que también le servirá de clave para descifrarlo.

```
[usuario@portatil:~]$ openssl rsautl -decrypt -inkey receptor/RSAkey.pem -in
  receptor/sessionkey_enc -out receptor/sessionkey
[usuario@portatil:~]$ cat receptor/sessionkey
402e49cf3b4789c0c1c700d300b7ed9b97e2bc8c760775b21e9ad4af035bd536837d5083df74d9c30
bb20f8aaf43fb79974c0ea2c82a99f43bb9101a47ff59100fc5fa002d1698433338f2c4d6250d35c5
1a68464dd95bb1a70023862bd96f979bfd2d7a167d7161150bf5c948afa5d79e253ac23dad24179e7
6d100141f021c
-aes-128-ecb

```

8. El receptor ya sabe cual es el algoritmo simétrico que se a utilizado para cifrar el mensaje, y también tiene la contraseña para poder descifrarlo.

```
[usuario@portatil:~]$ openssl enc -d -aes-128-ecb -pass file:sessionkey -nosalt -
  in receptor/imput.bin.enc.aes-128 -out receptor/imput.bin
[usuario@portatil:~]$ hexdump emisor/imput.bin
0000000 0000 0000 0000 0000 0000 0000 0000 0000
*
0000080
[usuario@portatil:~]$ hexdump receptor/imput.bin
0000000 0000 0000 0000 0000 0000 0000 0000 0000
*
0000080

```

Como se puede apreciar en el último paso los ficheros son identicos y el mensaje ha sido enviado con seguridad.

7. 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.

```
[usuario@portatil:~]$ openssl rsautl -encrypt -inkey emisor/RSApub.pem -pubin -in /tmp/
    imput.bin -out emisor/imput.bin.enc.rsa
[usuario@portatil:~]$ openssl rsautl -decrypt -inkey emisor/RSAkey.pem -in emisor/imput.
    bin.enc.rsa -out emisor/imput.bin
[usuario@portatil:~]$ hexdump /tmp/imput.bin
0000000 0000 0000 0000 0000 0000 0000 0000 0000
*
0000080
[usuario@portatil:~]$ hexdump emisor/imput.bin
0000000 0000 0000 0000 0000 0000 0000 0000 0000
*
0000080
```

Esta claro que después de realizar el proceso contrario de cifrar el mensaje, este sigue siendo el mismo, que es precisamente de lo que se trata.

8. 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.[2]

En mi implementación de OpenSSL no se encuentra ninguna de las dos curvas que se especifian en las transparencias de teoría, he podido darme cuenta que las que se utilizan para los servidores web [1] son *secp256k1* y *secp384r1* así que utilizaré la segunda

```
[usuario@portatil:~]$ openssl ecparam -list_curves
secp224r1 : NIST/SECG curve over a 224 bit prime field
secp256k1 : SECG curve over a 256 bit prime field
secp384r1 : NIST/SECG curve over a 384 bit prime field
secp521r1 : NIST/SECG curve over a 521 bit prime field
prime256v1: X9.62/SECG curve over a 256 bit prime field
[usuario@portatil:~]$ openssl ecparam -name secp384r1 -text -out emisor/stdECparam.key
[usuario@portatil:~]$ openssl ecparam -param_enc explicit -conv_form uncompressed -text -
    noout -no_seed -name secp384r1
Field Type: prime-field
Prime:
00:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:
ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:
ff:ff:fe:ff:ff:ff:ff:00:00:00:00:00:00:00:00:
ff:ff:ff:ff
A:
00:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:
ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:
ff:ff:fe:ff:ff:ff:ff:00:00:00:00:00:00:00:00:
ff:ff:ff:fc
B:
00:b3:31:2f:a7:e2:3e:e7:e4:98:8e:05:6b:e3:f8:
2d:19:18:1d:9c:6e:fe:81:41:12:03:14:08:8f:50:
13:87:5a:c6:56:39:8d:8a:2e:d1:9d:2a:85:c8:ed:
d3:ec:2a:ef
Generator (uncompressed):
04:aa:87:ca:22:be:8b:05:37:8e:b1:c7:1e:f3:20:
ad:74:6e:1d:3b:62:8b:a7:9b:98:59:f7:41:e0:82:
54:2a:38:55:02:f2:5d:bf:55:29:6c:3a:54:5e:38:
72:76:0a:b7:36:17:de:4a:96:26:2c:6f:5d:9e:98:
bf:92:92:dc:29:f8:f4:1d:bd:28:9a:14:7c:e9:da:
```

```
31:13:b5:f0:b8:c0:0a:60:b1:ce:1d:7e:81:9d:7a:
43:1d:7c:90:ea:0e:5f
Order:
00:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:
ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:c7:63:4d:81:f4:
37:2d:df:58:1a:0d:b2:48:b0:a7:7a:ec:ec:19:6a:
cc:c5:29:73
Cofactor:  1 (0x1)
```

9. 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.

```
[usuario@portatil:~]$ openssl ecparam -name secp384r1 -genkey -outform PEM -out emisor/
    CarlosECkey.pem
```

10. '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.

```
[usuario@portatil:~]$ cat emisor/CarlosECkey.pem | tail -n 6 > emisor/CarlosECpriv.pem
[usuario@portatil:~]$ openssl enc -aes-128-ecb -pass pass:mipass -nosalt -in emisor/
    CarlosECpriv.pem -out emisor/CarlosECpriv.pem.enc
[usuario@portatil:~]$ cat emisor/CarlosECpriv.pem
-----BEGIN EC PRIVATE KEY-----
MIGkAgEBBDAQXyxIfyjElNW9F/l3yHObQXWn3JrSHAehLJ5o6eJqMHD0fMmGBGwr
rM2cSOEiLtSgBwYFK4EEACKhZANiAATkeWYb92szqxW8kMMfT0Lxc+6THNVZawdd
RQfr77h8tjR9GezB1x5fEmodCfbk0s3qhJQISBItb791/vK2rQ8khUjTWWWGCLw8
gv0vZ102EFJx6CnPF6EHjY/i4yuqcPQ=
-----END EC PRIVATE KEY-----
[usuario@portatil:~]$ hexdump emisor/CarlosECpriv.pem.enc
0000000 8a70 2699 edd7 2d9f 1560 e7c2 2ac9 cb49
0000010 dd14 26d3 645b e437 416e 0c2e c3ae 8a58
0000020 18f0 421f 1101 8491 5a4c 8a18 d241 9ca1
0000030 6f19 0fca cbc4 bc5b 831f 6975 ba40 4e99
0000040 7dce 2951 401d b2b0 1cb8 b99d ff40 189c
0000050 b88c 7e3b 3e18 eb58 5f4b 971d 6284 dedf
0000060 dff8 d5a6 1c74 5895 73f5 c22b fcc9 fcab
0000070 b7a2 fd7a 13bb e30b a243 965b b837 c9c4
0000080 1604 7ece bcd7 9a09 1beb 4cda ac3a 9ca3
0000090 7518 cedb 6378 e002 7dfa 1fa7 18db e334
00000a0 f4f1 a329 ce65 9ce4 e4e3 2281 640a f86d
00000b0 3441 5093 4ed5 de9b ab23 ee50 7af9 b25b
00000c0 5728 452a dbd0 cd9e c33d 846c 00ba 301f
00000d0 8223 8e83 83ab 6369 0362 297a 622c 65d6
00000e0 4c69 b7e1 8320 e0ab eda5 2267 91e6 589b
00000f0 9380 30e2 e584 8152 0ab6 6377 5913 ca0d
0000100 5201 93e8 e015 74cb 58da d295 7e1e 46d5
0000110 8fb9 1235 3847 db83 b632 534a 7cd7 c9f0
0000120 eda3 8dff 37aa 320e 6a9f 82be 8ec9 4659
0000130
```

11. 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.

```
[usuario@portatil:~]$ openssl ec -in emisor/CarlosECkey.pem -pubout -outform PEM -out
    emisor/CarlosECpub.pem
read EC key
writing EC key
[usuario@portatil:~]$ cat emisor/CarlosECpub.pem
-----BEGIN PUBLIC KEY-----
MHYwEAYHKoZIzj0CAQYFK4EEACIDYgAE5HlmG/drM6sVvJDDH09C8XPukxzVWWsH
XUUH6++4fLY0fRnswdceXxJqHQn25NLN6oSUCEgSLW+/df7ytq0PJIVI011lhgi8
PILzr2ddNhBScegpzxehB42P4uMrqnD0
-----END PUBLIC KEY-----
```

Referencias

[1] Criptografía con curvas elípticas, Consultado el 4 de noviembre de 2018. http://www.criptored.upm.es/crypt4you/temas/ECC/leccion2/leccion2.html.

[2] Using sage to play with elliptic curves, Consultado el 4 de noviembre de 2018. https://www.johannes-bauer.com/compsci/ecc/#anchor37.