ΕΠΛ326: Εργαστήριο 4

Μη Συμμετρική Κρυπτογραφία

Εισαγωγή

Στο σημερινό εργαστήριο θα εφαρμόσουμε στην πράξη μη συμμετρική κρυπτογραφία και τις κρυπτογραφικές συναρτήσεις κατακερματισμού με τη βοήθεια του OpenSSL.

Εκτέλεση Εργαστηρίου

Βήμα 1

Εξερευνήστε τις εντολές ‘genrsa’ και ‘rsa’ στο OpenSSL.

```
$ openssl
OpenSSL> genrsa help
usage: genrsa [args] [numbits]
 -des            encrypt the generated key with DES in cbc mode
 -des3           encrypt the generated key with DES in ede cbc mode
(168 bit key)
 -seed
                 encrypt PEM output with cbc seed
 -aes128, -aes192, -aes256
                 encrypt PEM output with cbc aes
 -out file       output the key to 'file
 -passout arg    output file pass phrase source
 -f4             use F4 (0x10001) for the E value
 -3              use 3 for the E value
 -engine e       use engine e, possibly a hardware device.
 -rand file:file:...
                 load the file (or the files in the directory) into
                 the random number generator
error in genrsa

OpenSSL> rsa help
unknown option help
rsa [options] <infile >outfile
where options are
```

```
 -inform arg     input format - one of DER NET PEM
 -outform arg    output format - one of DER NET PEM
 -in arg         input file
 -sgckey         Use IIS SGC key format
 -passin arg     input file pass phrase source
 -out arg        output file
 -passout arg    output file pass phrase source
 -des            encrypt PEM output with cbc des
 -des3           encrypt PEM output with ede cbc des using 168 bit key
 -seed           encrypt PEM output with cbc seed
 -aes128, -aes192, -aes256
                 encrypt PEM output with cbc aes
 -text           print the key in text
 -noout          don't print key out
 -modulus        print the RSA key modulus
 -check          verify key consistency
 -pubin          expect a public key in input file
 -pubout         output a public key
 -engine e       use engine e, possibly a hardware device.
error in rsa
```

Βήμα 2

Κατασκευάστε ένα ζεύγος κλειδιών RSA.

```
$  openssl genrsa -des3 -out private.pem 2048
Generating RSA private key, 2048 bit long modulus
..............+++
..............+++
e is 65537 (0x10001)
Enter pass phrase for private.pem:
Verifying - Enter pass phrase for private.pem:
(elathan@akasaka-2:~/ucy/teaching/epl326/labs/4)$ ls -l
total 16
-rw-r--r--@ 1 elathan  staff  4074 Aug 25 15:19 lab-asymmetric-
encryption
-rw-r--r--@ 1 elathan  staff  1743 Aug 25 15:20 private.pem
```

Το αρχείο *private.pem* έχει αποθηκεύσει το ζεύγος, το οποίο το προστατεύει με ένα κλειδί και τον αλγόριθμο DES3. Το κλειδί συνδέεται με το passphrase που εσείς δίνετε.

Μπορείτε να αφαιρέσετε το passphrase

```
$ openssl rsa -in private.pem -out privatekey.pem
```

Μπορείτε αν θέλετε να επιλέξετε να μην προστατευθεί το κλειδί με κάποιο passphrase (όχι πολύ ασφαλής επιλογή).

```
$  openssl genrsa -out private.pem 2048
Generating RSA private key, 2048 bit long modulus
.....................................................................+++
..................................+++
e is 65537 (0x10001)
```

Βήμα 3

Δείτε πώς είναι το ζεύγος κλειδιών.

```
$ cat private.pem
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEA5uyu4DkL9cp1yi+lnEHc7jqHCTzlO+MdZgWLcntw3TfJyHrA
f7gVX2s7KalUkU/uKP3Zp3g8dcz4vjjS8GE8vonsf6zTO6/susf8FhhtuHhyrCH9
hAcnUC2jzVY6xaNJk5FbrbQej5/09PfpszVfGnKMJqglgDzxj9dVJDbawtMy8yNv
uX8a35IVRaN21S6RDfxBHyKyVlGxTEIDqOGm0F+2fiYqT40FOf7dZKY1Mj2k1ZDR
B1NvLFn2dpwmFVf5BVTHAs8z1WNzV+LrzsqlUDQK70lKAswRcsKn/y6D9CHmslj/
1qCGYzghMrDb7ZlmgV9VazEQbIkwOeaYxJlNgQIDAQABAoIBAGqc4Uvx6nyq4WDc
SJhwrABskwWH0MmqXFtKugu7w+rOI+HkElWk9nEcP69HVxzTiz/vmvo/f6JfZIPm
FjUWn4XUhZSYhNlsql29kLvewWFGcqLX+oRnDRZuS860PeQbNA/mLy5uFyH2FpC2
mHW9cU/zjJMYVF/1Qw1GvC9BCBSjrq2D2s2R5AJo5DJ3ExRPstIi0Egzn2W/LSGt
qHZLUrA2t2e2iyqQvJf0SnjTfgR6mIVSCounHF6H+spw6k18XVCg/qZxMLjvMhzF
TlHGjE7sIxLGbtl31MPodfyedjr5OPjnnXnQW0n2osWI80uM1zQqW9/DHcjxSaHA
5bh0XgUCgYEA+elrRGap5oW+usyA6/OZadHENnSJvM1pfFMpGcRyJgKfCzpIXeXY
WSBNJ800OFclxoAUUWdpuB4uqIYgB7OT0OrIKPSwDp9c/1VKoAs2232r0DxxeIX7
zsgODCoGOX0P0Mhpsl8a3675di8NdPowlHKhvJdQSS8B26N8pxQ/UzcCgYEA7IzZ
gsEhjwx13UQ37UtnhdQ2duOXMba1kduFHB3I9+Tl6I9yl5DFNBPMRdbz5Z9hO4fq
rUgxmShgv7nQNG1vwudMnf4Dd/CwcJNHyB7yeRoU8RcGEYl7a+YU9A2jY5tqJYgN
/s+pAhqdABQDLAZ0JKtWd6J70Zq5bBmT9OjTsQcCgYEA9UZbKfHctEv1qp5Ftc4K
dqoS+iMcf/nWmtd/kjOUmKuf7sQP/GA2GxsxvmuO4fDeOVjBGDf3sWWmA25IB2b/
H18efdvQ4zkTa+II8NSwzXEk7KY188UEI6mC5/tiyo9ss0vPOOpQlNLFMXHysilO
/pMXG+IiQQFqwGdPUyfzErcCgYEA4xujpUbZembDLonVjCzuGm4RUHnX6nH5xyzU
kpcCIQaRuYMnyN7EhuhPIeprRdWMW8R1iqKag+phYRH8FKnROHYreDCtkdTzAVJW
A4u6zow8+O9a8zFk3sYK7fm+Gd7XvWrvzOoA4LU+cRnFpdoYXjBmFvxYFniPd3rj
M6rpu7cCgYEA4sNU0ZK8BPZn+ZCQNcK4uN22R7GxSiSuerBCvvPwBNiDGAeOnAym
zlFMJNKQg0P+lSBpae7IPuRDy/xZzrlM0B/fMLul6USywDUZH1GKCbGxLxS0Nedq
```

```
+e9ZtoNGDQytU8msQFzsOkwrxpmrrz6Ui79Tu11b+rxpP/u/ZIYtRpw=
-----END RSA PRIVATE KEY-----
```

Δείτε τις παραμέτρους του ζεύγους κλειδιών (π.χ., τους πρώτους αριθμούς, q και p, που έχουν χρησιμοποιηθεί).

```
$ openssl rsa -noout -text -in private.pem
Private-Key: (2048 bit)
modulus:
    00:e6:ec:ae:e0:39:0b:f5:ca:75:ca:2f:a5:9c:41:
    dc:ee:3a:87:09:3c:e5:3b:e3:1d:66:05:8b:72:7b:
    70:dd:37:c9:c8:7a:c0:7f:b8:15:5f:6b:3b:29:a9:
    54:91:4f:ee:28:fd:d9:a7:78:3c:75:cc:f8:be:38:
    d2:f0:61:3c:be:89:ec:7f:ac:d3:3b:af:ec:ba:c7:
    fc:16:18:6d:b8:78:72:ac:21:fd:84:07:27:50:2d:
    a3:cd:56:3a:c5:a3:49:93:91:5b:ad:b4:1e:8f:9f:
    f4:f4:f7:e9:b3:35:5f:1a:72:8c:26:a8:25:80:3c:
    f1:8f:d7:55:24:36:da:c2:d3:32:f3:23:6f:b9:7f:
    1a:df:92:15:45:a3:76:d5:2e:91:0d:fc:41:1f:22:
    b2:56:51:b1:4c:42:03:a8:e1:a6:d0:5f:b6:7e:26:
    2a:4f:8d:05:39:fe:dd:64:a6:35:32:3d:a4:d5:90:
    d1:07:53:6f:2c:59:f6:76:9c:26:15:57:f9:05:54:
    c7:02:cf:33:d5:63:73:57:e2:eb:ce:ca:a5:50:34:
    0a:ef:49:4a:02:cc:11:72:c2:a7:ff:2e:83:f4:21:
    e6:b2:58:ff:d6:a0:86:63:38:21:32:b0:db:ed:99:
    66:81:5f:55:6b:31:10:6c:89:30:39:e6:98:c4:99:
    4d:81
publicExponent: 65537 (0x10001)
privateExponent:
    6a:9c:e1:4b:f1:ea:7c:aa:e1:60:dc:48:98:70:ac:
    00:6c:93:05:87:d0:c9:aa:5c:5b:4a:ba:0b:bb:c3:
    ea:ce:23:e1:e4:12:55:a4:f6:71:1c:3f:af:47:57:
    1c:d3:8b:3f:ef:9a:fa:3f:7f:a2:5f:64:83:e6:16:
    35:16:9f:85:d4:85:94:98:84:d9:6c:aa:5d:bd:90:
    bb:de:c1:61:46:72:a2:d7:fa:84:67:0d:16:6e:4b:
    ce:b4:3d:e4:1b:34:0f:e6:2f:2e:6e:17:21:f6:16:
    90:b6:98:75:bd:71:4f:f3:8c:93:18:54:5f:f5:43:
    0d:46:bc:2f:41:08:14:a3:ae:ad:83:da:cd:91:e4:
    02:68:e4:32:77:13:14:4f:b2:d2:22:d0:48:33:9f:
    65:bf:2d:21:ad:a8:76:4b:52:b0:36:b7:67:b6:8b:
    2a:90:bc:97:f4:4a:78:d3:7e:04:7a:98:85:52:0a:
    8b:a7:1c:5e:87:fa:ca:70:ea:4d:7c:5d:50:a0:fe:
    a6:71:30:b8:ef:32:1c:c5:4e:51:c6:8c:4e:ec:23:
    12:c6:6e:d9:77:d4:c3:e8:75:fc:9e:76:3a:f9:38:
    f8:e7:9d:79:d0:5b:49:f6:a2:c5:88:f3:4b:8c:d7:
```

```
    34:2a:5b:df:c3:1d:c8:f1:49:a1:c0:e5:b8:74:5e:
    05
prime1:
    00:f9:e9:6b:44:66:a9:e6:85:be:ba:cc:80:eb:f3:
    99:69:d1:c4:36:74:89:bc:cd:69:7c:53:29:19:c4:
    72:26:02:9f:0b:3a:48:5d:e5:d8:59:20:4d:27:cd:
    34:38:57:25:c6:80:14:51:67:69:b8:1e:2e:a8:86:
    20:07:b3:93:d0:ea:c8:28:f4:b0:0e:9f:5c:ff:55:
    4a:a0:0b:36:db:7d:ab:d0:3c:71:78:85:fb:ce:c8:
    0e:0c:2a:06:39:7d:0f:d0:c8:69:b2:5f:1a:df:ae:
    f9:76:2f:0d:74:fa:30:94:72:a1:bc:97:50:49:2f:
    01:db:a3:7c:a7:14:3f:53:37
prime2:
    00:ec:8c:d9:82:c1:21:8f:0c:75:dd:44:37:ed:4b:
    67:85:d4:36:76:e3:97:31:b6:b5:91:db:85:1c:1d:
    c8:f7:e4:e5:e8:8f:72:97:90:c5:34:13:cc:45:d6:
    f3:e5:9f:61:3b:87:ea:ad:48:31:99:28:60:bf:b9:
    d0:34:6d:6f:c2:e7:4c:9d:fe:03:77:f0:b0:70:93:
    47:c8:1e:f2:79:1a:14:f1:17:06:11:89:7b:6b:e6:
    14:f4:0d:a3:63:9b:6a:25:88:0d:fe:cf:a9:02:1a:
    9d:00:14:03:2c:06:74:24:ab:56:77:a2:7b:d1:9a:
    b9:6c:19:93:f4:e8:d3:b1:07
exponent1:
    00:f5:46:5b:29:f1:dc:b4:4b:f5:aa:9e:45:b5:ce:
    0a:76:aa:12:fa:23:1c:7f:f9:d6:9a:d7:7f:92:33:
    94:98:ab:9f:ee:c4:0f:fc:60:36:1b:1b:31:be:6b:
    8e:e1:f0:de:39:58:c1:18:37:f7:b1:65:a6:03:6e:
    48:07:66:ff:1f:5f:1e:7d:db:d0:e3:39:13:6b:e2:
    08:f0:d4:b0:cd:71:24:ec:a6:35:f3:c5:04:23:a9:
    82:e7:fb:62:ca:8f:6c:b3:4b:cf:38:ea:50:94:d2:
    c5:31:71:f2:b2:29:4e:fe:93:17:1b:e2:22:41:01:
    6a:c0:67:4f:53:27:f3:12:b7
exponent2:
    00:e3:1b:a3:a5:46:d9:7a:66:c3:2e:89:d5:8c:2c:
    ee:1a:6e:11:50:79:d7:ea:71:f9:c7:2c:d4:92:97:
    02:21:06:91:b9:83:27:c8:de:c4:86:e8:4f:21:ea:
    6b:45:d5:8c:5b:c4:75:8a:a2:9a:83:ea:61:61:11:
    fc:14:a9:d1:38:76:2b:78:30:ad:91:d4:f3:01:52:
    56:03:8b:ba:ce:8c:3c:f8:ef:5a:f3:31:64:de:c6:
    0a:ed:f9:be:19:de:d7:bd:6a:ef:cc:ea:00:e0:b5:
    3e:71:19:c5:a5:da:18:5e:30:66:16:fc:58:16:78:
    8f:77:7a:e3:33:aa:e9:bb:b7
coefficient:
    00:e2:c3:54:d1:92:bc:04:f6:67:f9:90:90:35:c2:
    b8:b8:dd:b6:47:b1:b1:4a:24:ae:7a:b0:42:be:f3:
```

```
f0:04:d8:83:18:07:8e:9c:0c:a6:ce:51:4c:24:d2:
90:83:43:fe:95:20:69:69:ee:c8:3e:e4:43:cb:fc:
59:ce:b9:4c:d0:1f:df:30:bb:a5:e9:44:b2:c0:35:
19:1f:51:8a:09:b1:b1:2f:14:b4:35:e7:6a:f9:ef:
59:b6:83:46:0d:0c:ad:53:c9:ac:40:5c:ec:3a:4c:
2b:c6:99:ab:af:3e:94:8b:bf:53:bb:5d:5b:fa:bc:
69:3f:fb:bf:64:86:2d:46:9c
```

Δημιουργήστε ένα ζεύγος κλειδιών RSA.

```
$  openssl genrsa -des3 -out private.pem 2048
```

Δημιουργήστε ένα αρχείο (sender.txt) με το κείμενο “What is my final exam score”.

$ gedit sender.txt &

Εξάγετε το δημόσιο κλειδί από το ιδιωτικό κλειδί.

```
$ openssl rsa -in private.pem -pubout -out public.pem
```

Δείτε πώς είναι το δημόσιο κλειδί, public key.

$cat public.pem

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAyZHVHCicTmTseHq7OQIL
jlMJ67LAv31sloiuzWb8yzgKcTE09DgTEzuw+s8kB64m5nXMfBJGCm1oWYDZh5VW
BME1R/PMvBHm2DP7UhTArCU9MpaPedXmJ6A9mClmGdLJULSmjHbtAAHTmbbL6WXW
bc7pnNkk+5o/4YRk0F1xXRDq2QS7BfoPL+qmucM/vGYm0kZ/kis5//TIryV4AkWv
/6q4WDtBZAxP2DLpyEm338KlIOajgUFH+m8ivO1aRxATLZ/Oad40+gvkY6Er1cCB
fri0mCAIy7XvW0E54j31nczuLTifBBrGD7gZcicBYTRPfhx5OTvASH17GA1Gx71a
KwIDAQAB
-----END PUBLIC KEY-----
```

Κρυπτογραφήστε το sender.txt με το ιδιωτικό κλειδί.

```
$openssl rsautl -sign -inkey private.pem -in sender.txt >
sender_enc_prv
```

Αποκρυπτογραφήστε το sender_enc_prv με το δημόσιο κλειδί.

```
$openssl rsautl -verify -inkey  public.pem  -pubin -in sender_enc_prv
```

ή

```
$openssl rsautl -verify -inkey private.pem -in sender_enc_prv
```

Δημιουργήστε ένα αρχείο (rcv.txt) με το κείμενο “Your final exam score was 100”.

```
$ gedit rcv.txt &
```

Κρυπτογραφήστε το rcv.txt με το δημόσιο κλειδί.

```
$openssl rsautl -encrypt -inkey private.pem -in rcv.txt -out
rcvpub.enc
```

ή

```
$openssl rsautl  -encrypt -inkey  public.pem  -pubin -in rcv.txt
-out rcvpub.enc2
```

Αποκρυπτογραφήστε το `rcvpub.enc` με το ιδιωτικό κλειδί.

```
$ openssl rsautl -decrypt -inkey private.pem -in rcvpub.enc -out
rcvtext.dec
```

Βήμα 4

Εξερευνήστε τις κρυπτογραφικές συναρτήσεις κατακερματισμού που χρησιμοποιεί το OpenSSL χρησιμοποιώντας την εντολή ‘dgst’.

```
$ openssl dgst -help
unknown option '-help'
options are
-c              to output the digest with separating colons
-d              to output debug info
-hex            output as hex dump
-binary         output in binary form
-sign   file    sign digest using private key in file
-verify file    verify a signature using public key in file
-prverify file  verify a signature using private key in file
-keyform arg    key file format (PEM or ENGINE)
-signature file signature to verify
-binary         output in binary form
-hmac key       create hashed MAC with key
-engine e       use engine e, possibly a hardware device.
-md5            to use the md5 message digest algorithm (default)
-md4            to use the md4 message digest algorithm
-md2            to use the md2 message digest algorithm
-sha1           to use the sha1 message digest algorithm
-sha            to use the sha message digest algorithm
```

```
-sha224         to use the sha224 message digest algorithm
-sha256         to use the sha256 message digest algorithm
-sha384         to use the sha384 message digest algorithm
-sha512         to use the sha512 message digest algorithm
-mdc2           to use the mdc2 message digest algorithm
-ripemd160      to use the ripemd160 message digest algorithm
```

Βήμα 5

Χρησιμοποιήστε μερικές.

```
$ openssl md5 /etc/passwd
MD5(/etc/passwd)= 5e7f80888f3d491c4963881364048c24

$ openssl dgst -sha256 /etc/passwd
SHA256(/etc/passwd)=
39c487734fed185cf16217552ed8b451525c240e13d41001b3782b46fdcf4708
```

Συνεχίστε με μερικά ακόμα παραδείγματα της επιλογής σας.

Βήμα 6

Υπολογίστε το digest δικών σας δεδομένων.

```
$ echo "Hello World" | openssl dgst -sha256
d2a84f4b8b650937ec8f73cd8be2c74add5a911ba64df27458ed8229da804a26
```

Δοκιμάστε με διαφορετικά μηνύματα και διαφορετικούς αλγορίθμους

```
$ for i in {1..1000}; do echo $RANDOM; done > random.data
```

Το αρχείο αυτό περιέχει τυχαίους αριθμούς.

```
$ openssl dgst -sha256 ./random.data
SHA256(./random.data)=
f4e709b87daf4ddea6278669224b8dd2f6b4b15ebf3d65c404d97b2fab2dfcf9
```

Αλλάξτε έναν αριθμό με τη βοήθεια ενός editor και υπολογίστε ξανά το digest.

```
$ openssl dgst -sha256 ./random.data
SHA256(./random.data)=
614a2a0303e1bad7b1e6c57bf5d68bc6c4213c0ca26e4935d579c6aa7a3cf16b
```

Δείτε πόσο διαφορετικά είναι τα digests.

Βήμα 7

Δείτε τα αρχεία *shattered-1.pdf* και *shattered-2.pdf.* Υπολογίστε τα digest με τη βοήθεια του αλγορίθμου SHA256 και SHA1. Τί παρατηρείτε;

```
$ openssl dgst -sha256 shattered-1.pdf
SHA256(shattered-1.pdf)=
2bb787a73e37352f92383abe7e2902936d1059ad9f1ba6daaa9c1e58ee6970d0
$ openssl dgst -sha256 shattered-2.pdf
SHA256(shattered-2.pdf)=
d4488775d29bdef7993367d541064dbdda50d383f89f0aa13a6ff2e0894ba5ff

$ openssl dgst -sha1 shattered-1.pdf
SHA1(shattered-1.pdf)= 38762cf7f55934b34d179ae6a4c80cadccbb7f0a
$ openssl dgst -sha1 shattered-2.pdf
SHA1(shattered-2.pdf)= 38762cf7f55934b34d179ae6a4c80cadccbb7f0a
```

Βήμα 8

Υπογράψτε ένα αρχείο. Πρώτα πρέπει να κατασκευάσετε το digest του αρχείου με τη βοήθεια μιας κρυπτογραφικής συνάρτησης κατακερματισμού (π.χ., SHA256).

```
$ openssl dgst -sha256 /etc/passwd > etc.passwd.hash
$ cat etc.passwd.hash
SHA256(/etc/passwd)=
39c487734fed185cf16217552ed8b451525c240e13d41001b3782b46fdcf4708
```

Και μετά να κρυπτογραφήσετε το digest χρησιμοποιώντας το ιδιωτικό κλειδί του RSA.

```
$ openssl rsautl -sign -inkey private.pem -in etc.passwd.hash >
signature
$ file signature
signature: data
```

Επιβεβαιώστε την υπογραφή με το δημόσιο κλειδί του RSA.

```
$ openssl rsautl -verify -inkey private.pem -in signature
```

(extract public key from private)

```
openssl rsa -in private.pem -pubout -out public.pem
```

Επιβεβαιώστε την υπογραφή με το δημόσιο κλειδί του RSA.

```
$ openssl rsautl -inkey  public.pem -pubin -in signature
```

Βημα 9

Κρυπτογραφήστε ένα digest με το δημόσιο κλειδί και αποκρυπτογραφήστε το με το ιδιωτικό.

```
$ openssl rsautl -encrypt -inkey private.pem -in etc.passwd.hash -out
file.enc
```

Και αποκρυπτογραφήσετε το με το ιδιωτικό.

```
$ openssl rsautl -decrypt -inkey private.pem -in file.enc -out
file.dec
```

Βήμα 10:

Τα δεδομένα που βλέπετε πιο κάτω είναι τα digest των κωδικών κάποιων χρηστών που το user id τους αναγράφετε στα αριστερά. Δηλαδή ο χρηστής elias έχει κωδικό που το digest του είναι ca……ca9. Επίσης γνωρίζετε ότι το hash function που χρησιμοποιήθηκε για να παραχθούν τα digests είναι το md5. Μπορείτε να βρείτε τους κωδικούς των πιο κάτω χρηστών από τα digests.

Elias: ca1c76ae2638aa4fb9a708b167386ca9
Yiannos: b26e2722e8b42f89c66c97f332f3cb56
User12: e7df7cd2ca07f4f1ab415d457a6e1c13
Elleni1999: f95f8943f6dcf7b3c1c8c2cab5455f8b
User15: a86850deb2742ec3cb41518e26aa2d89

```
modulus         INTEGER,  -- n
 publicExponent   INTEGER,  -- e
 privateExponent  INTEGER,  -- d
 prime1          INTEGER,  -- p
 prime2          INTEGER,  -- q
 exponent1       INTEGER,  -- d mod (p-1)
 exponent2       INTEGER,  -- d mod (q-1)
 coefficient     INTEGER,  -- (inverse of q) mod p
```