OpenSSL

OpenSSL is a software library for applications that provide secure communications over computer networks against eavesdropping or need to identify the party at the other end. It is widely used by Internet servers, including the majority of HTTPS websites.

1. Check version openssl

```
[nikita@Air ~$ openssl version
LibreSSL 3.3.6
```

2. Detail check version openssl

```
[nikita@Air ~$ openssl version -a
LibreSSL 3.3.6
built on: date not available
platform: information not available
options:  bn(64,64) rc4(ptr,int) des(idx,cisc,16,int) blowfish(idx)
compiler: information not available
OPENSSLDIR: "/private/etc/ssl"
```

3. Available Commands

```
[nikita@Air ~$ openssl help
openssl:Error: 'help' is an invalid command.

Standard commands
asn1parse         ca                certhash          ciphers
cms               crl               crl2pkcs7         dgst
dh                dhparam           dsa               dsaparam
ec                ecparam           enc               errstr
gendh             gendsa            genpkey           genrsa
nseq              ocsp              passwd            pkcs12
pkcs7             pkcs8             pkey              pkeyparam
pkeyutl           prime             rand              req
rsa               rsautl            s_client          s_server
s_time            sess_id           smime             speed
spkac             ts                verify            version
x509

Message Digest commands (see the `dgst' command for more details)
gost-mac          md4               md5               md_gost94
ripemd160         sha1              sha224            sha256
sha384            sha512            sm3               sm3WithRSAEncryption
streebog256       streebog512       whirlpool

Cipher commands (see the `enc' command for more details)
aes-128-cbc       aes-128-ecb       aes-192-cbc       aes-192-ecb
aes-256-cbc       aes-256-ecb       base64            bf
bf-cbc            bf-cfb            bf-ecb            bf-ofb
camellia-128-cbc  camellia-128-ecb  camellia-192-cbc  camellia-192-ecb
camellia-256-cbc  camellia-256-ecb  cast              cast-cbc
cast5-cbc         cast5-cfb         cast5-ecb         cast5-ofb
chacha            des               des-cbc           des-cfb
des-ecb           des-ede           des-ede-cbc       des-ede-cfb
des-ede-ofb       des-ede3          des-ede3-cbc      des-ede3-cfb
des-ede3-ofb      des-ofb           des3              desx
rc2               rc2-40-cbc        rc2-64-cbc        rc2-cbc
rc2-cfb           rc2-ecb           rc2-ofb           rc4
rc4-40            sm4               sm4-cbc           sm4-cfb
sm4-ecb           sm4-ofb
```

4. Displaying a list of available commands with subcommands

```
[nikita@Air ~$ openssl rsa help
unknown option 'help'
usage: rsa [-ciphername] [-check] [-in file] [-inform fmt]
    [-modulus] [-noout] [-out file] [-outform fmt] [-passin src]
    [-passout src] [-pubin] [-pubout] [-sgckey] [-text]

 -check             Check consistency of RSA private key
 -in file           Input file (default stdin)
 -inform format     Input format (DER, NET or PEM (default))
 -modulus           Print the RSA key modulus
 -noout             Do not print encoded version of the key
 -out file          Output file (default stdout)
 -outform format    Output format (DER, NET or PEM (default PEM))
 -passin src        Input file passphrase source
 -passout src       Output file passphrase source
 -pubin             Expect a public key (default private key)
 -pubout            Output a public key (default private key)
 -sgckey            Use modified NET algorithm for IIS and SGC keys
 -text              Print in plain text in addition to encoded
```

5. Displaying all ciphers

```
nikita@Air ~$ openssl ciphers

[AEAD-CHACHA20-POLY1305-SHA256:AEAD-AES256-GCM-SHA384:AEAD-AES128-GCM-SHA256:ECDHE-ECDSA-CHACHA20
-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-CHACHA20-POLY1305:ECDHE-RSA-AES256-GCM-SHA384:ECDH
E-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA
:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-SHA256:DHE-RSA-AES256-SHA:GOST2
012256-GOST89-GOST89:DHE-RSA-CAMELLIA256-SHA256:DHE-RSA-CAMELLIA256-SHA:GOST2001-GOST89-GOST89:A
ES256-GCM-SHA384:AES256-SHA256:AES256-SHA:CAMELLIA256-SHA256:CAMELLIA256-SHA:ECDHE-RSA-AES128-GC
M-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-R
SA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-SHA256:DHE-RSA-AES
128-SHA:DHE-RSA-CAMELLIA128-SHA256:DHE-RSA-CAMELLIA128-SHA:AES128-GCM-SHA256:AES128-SHA256:AES12
8-SHA:CAMELLIA128-SHA256:CAMELLIA128-SHA:ECDHE-RSA-RC4-SHA:ECDHE-ECDSA-RC4-SHA:RC4-SHA:RC4-MD5:E
CDHE-RSA-DES-CBC3-SHA:ECDHE-ECDSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:DES-CBC3-SHA
```

6. Check speed for operation

```
nikita@Air ~$ openssl speed ecdh

Doing 160 bit  ecdh's for 10s: 33083 160-bit ECDH ops in 9.98s
Doing 192 bit  ecdh's for 10s: 31937 192-bit ECDH ops in 9.98s
Doing 224 bit  ecdh's for 10s: 23740 224-bit ECDH ops in 9.96s
Doing 256 bit  ecdh's for 10s: 26337 256-bit ECDH ops in 9.95s
Doing 384 bit  ecdh's for 10s: 10481 384-bit ECDH ops in 9.98s
Doing 521 bit  ecdh's for 10s: 5490 521-bit ECDH ops in 9.98s
```

7. Check Connection

```
[nikita@Air ~$ openssl s_time -connect google.com:443                    ]
Collecting connection statistics for 30 seconds
**********************************************************************************
**********************************************************************************
******************

178 connections in 0.26s; 672.65 connections/user sec, bytes read 0
178 connections in 30 real seconds, 0 bytes read per connection


Now timing with session id reuse.
starting
**********************************************************************************
**********************************************************************************
```

```
**********************************************************************************
*****************************************

40832285 connections in 16.24s; 2514295.21 connections/user sec, bytes read 0
40832285 connections in 30 real seconds, 0 bytes read per connection
```

8. Encryption and decryption of data using aes_256_cbc

Creating a document -> with encrypted information

```
[nikita@Air ~/5 курс/криптография/10 Openssl$ nano doc
[nikita@Air ~/5 курс/криптография/10 Openssl$ cat doc
Data about Nikita

[nikita@Air ~/5 курс/криптография/10 Openssl$ openssl
OpenSSL> enc -e -aes-256-cbc -in ./doc -out ./enc_doc -pbkdf2
[enter aes-256-cbc encryption password:
[Verifying - enter aes-256-cbc encryption password:
OpenSSL> q
[nikita@Air ~/5 курс/криптография/10 Openssl$ cat enc_doc
Salted__?e-??1?N?'????w??????6??N?+?8fм.?%
```

Creating a document -> with decrypted information

```
nikita@Air ~/5 курс/криптография/10 Openssl$ openssl
OpenSSL> enc -d -aes-256-cbc -in ./enc_doc -out ./new_doc -pbkdf2
[enter aes-256-cbc decryption password:
OpenSSL> q
[nikita@Air ~/5 курс/криптография/10 Openssl$ cat new_doc
Data about Nikita
```

9. Check speed for algorithm aes-256-cbc

```
[nikita@Air ~$ openssl speed aes-256-cbc
Doing aes-256 cbc for 3s on 16 size blocks: 51952036 aes-256 cbc's in 3.00s
Doing aes-256 cbc for 3s on 64 size blocks: 11820155 aes-256 cbc's in 3.00s
Doing aes-256 cbc for 3s on 256 size blocks: 2935026 aes-256 cbc's in 3.00s
Doing aes-256 cbc for 3s on 1024 size blocks: 741872 aes-256 cbc's in 3.00s
Doing aes-256 cbc for 3s on 8192 size blocks: 93128 aes-256 cbc's in 3.00s
```

10. Encryption and decryption data with des-cbc

```
nikita@Air ~/5 курс/криптография/10 Openssl/des-cbc$ touch doc
nikita@Air ~/5 курс/криптография/10 Openssl/des-cbc$ nano doc
nikita@Air ~/5 курс/криптография/10 Openssl/des-cbc$ cat doc
Data about Nikita

nikita@Air ~/5 курс/криптография/10 Openssl/des-cbc$ openssl
OpenSSL> enc -e -des-cbc -in ./doc -out ./enc_file -pass pass:1 -nosalt -pbkdf2
OpenSSL> q
nikita@Air ~/5 курс/криптография/10 Openssl/des-cbc$ cat enc_file
?2P=?)?dl%
```

Decryption:

```
nikita@Air ~/5 курс/криптография/10 Openssl/des-cbc$ openssl
OpenSSL> enc -d -des-cbc -in ./enc_file -out ./new_file -pass pass:1 -nosalt -pbkdf2
OpenSSL> q
nikita@Air ~/5 курс/криптография/10 Openssl/des-cbc$ cat new_file
Data about Nikita
```

13. Encryption and decryption of data using a foreign key

```
nikita@Air ~/5 курс/криптография/10 Openssl/des_with_keyfile$ touch file
nikita@Air ~/5 курс/криптография/10 Openssl/des_with_keyfile$ ls
file
nikita@Air ~/5 курс/криптография/10 Openssl/des_with_keyfile$ nano file
nikita@Air ~/5 курс/криптография/10 Openssl/des_with_keyfile$ cat file
Data
nikita@Air ~/5 курс/криптография/10 Openssl/des_with_keyfile$ openssl rand -hex 10 > key_file
nikita@Air ~/5 курс/криптография/10 Openssl/des_with_keyfile$ cat key_file
bee1283a31dcc0205afe
nikita@Air ~/5 курс/криптография/10 Openssl/des_with_keyfile$ openssl
OpenSSL> enc -e -des-cbc -in ./file -out ./enc_file -pass file:key_file -nosalt -pbkdf2
OpenSSL> q
nikita@Air ~/5 курс/криптография/10 Openssl/des_with_keyfile$ cat enc_file
??}?Иh%
```

Decryption:

```
nikita@Air ~/5 курс/криптография/10 Openssl/des_with_keyfile$ openssl
OpenSSL> enc -d -des-cbc -in ./enc_file -out ./new_file -pass file:key_file -nosalt -pbkdf2
OpenSSL> q
nikita@Air ~/5 курс/криптография/10 Openssl/des_with_keyfile$ cat ./new_file
Data
```

14. Generation key for using RSA

```
nikita@Air ~/5 курс/криптография/10 Openssl/rsa$ openssl
OpenSSL> genrsa -out priv.pem 1024
Generating RSA private key, 1024 bit long modulus
............+++++
...+++++
e is 65537 (0x10001)
OpenSSL> q
nikita@Air ~/5 курс/криптография/10 Openssl/rsa$ cat priv.pem
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCr9/O4XUfGsBfLF2wysjVw580mhxpJV07oqx9k4xmwflpfoj57
ACAVkVRLt8YrDnUY8hjSYcH61/pGfC/Tbypvb/gdz6/GuMGlYRVG+ZV4DQm5Fr55
TY1vwvcBw7s1dGgZpUh9CdQYtV59c8/qxRyeJTz6l8OgJuWT5VkjqThdUQIDAQAB
AoGBAITZn4HNJjb7MNZneOHNRHW9zYG3G3qPJVmfYThqLR1o7140E7qoBSFETW/V
QBVhLNshpEVa2iV0Fle/dp8LoSaDY5WXPV3rtFz1N4roh1gHB16tQC9lFXtkOxuH
1bk4bE7MAwEnItaIqh+xVQzyRREtfo6pIKE5oc+mCDNWncxBAkEA2TATGqFYXg5Z
Kbj5fDAjIlwec1JsnMY039sG/wBPFJKrBE4XMPSbSix/uS2JuZuVfxl7be5AlRRS
XQBMezSPmQJBAMqzLy9Ya6DoLJIHeynxojTxxARIcbsk+7Q1cI/8TyB5dyRlC+DK
lqXHxQy1/xiR1sZjwKfGzyM1a/JHeFALLnkCQF7IbT3s9fyuNFlSAYciFuMy4pMf
gIj2szKSir4Uq25mB75sOhDCX1cCjocJCZb4AEkknMo/9O1angs6SyZlrNkCQCis
wjgu9xIH5QG8rJNjIJidNydg3imepz+nRZovGDW9ChJskHCgVpCXwMvnLXRht4aB
9+Py+hfnnWy1uMffJokCQQCHEJ7A7D2ZtaSSJthxKY6oUyL868bwmwXxUv6D1aLu
h9JHbpMopXbtAbGI1qV6mJ5T42PHCv6omxIOh4Cwx7sq
-----END RSA PRIVATE KEY-----
```

```
[nikita@Air ~/5 курс/криптография/10 Openssl/rsa$ cat pub.pem
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCr9/O4XUfGsBfLF2wysjVw580m
hxpJV07oqx9k4xmwflpfoj57ACAVkVRLt8YrDnUY8hjSYcH61/pGfC/Tbypvb/gd
z6/GuMGlYRVG+ZV4DQm5Fr55TY1vwvcBw7s1dGgZpUh9CdQYtV59c8/qxRyeJTz6
l8OgJuWT5VkjqThdUQIDAQAB
-----END PUBLIC KEY-----
```

15. Encryption and decryption with RSA

```
[nikita@Air ~/5 курс/криптография/10 Openssl/rsa$ cat file
Data
[nikita@Air ~/5 курс/криптография/10 Openssl/rsa$ openssl
OpenSSL> rsautl -encrypt -in ./file -out ./enc_file -inkey pub.pem -pubin
OpenSSL> q
[nikita@Air ~/5 курс/криптография/10 Openssl/rsa$ cat ./enc_file
????????????>B?46?????b_?
                        !n??rg?}K?*???pOA?#q!?5???K®(?{?#4V??_@?)?=5?x,??u?`gnarW??O%}?3???]
?*Et6"%
```

Decryption:

```
nikita@Air ~/5 курс/криптография/10 Openssl/rsa$ openssl
OpenSSL> rsautl -decrypt -in ./enc_file -out ./new_file -inkey priv.pem
OpenSSL> q
[nikita@Air ~/5 курс/криптография/10 Openssl/rsa$ cat new_file
Data
```

Using algorithms from openssl to check for data changes.

1. Generating an Encryption Key for the DES Algorithm

```
[nikita@Air ~/5 курс/криптография$ openssl
OpenSSL> genrsa -out ./priv.pem -des 1024
Generating RSA private key, 1024 bit long modulus
.....................................................+++++
.....................+++++
e is 65537 (0x10001)
[Enter pass phrase for ./priv.pem:
[Verifying - Enter pass phrase for ./priv.pem:
OpenSSL> q
[nikita@Air ~/5 курс/криптография$ cat priv.pem
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-CBC,CDDA5990857F4C50

07FRag4k89+x7/SpBnCEwx5d9WQF6YTFvg5W5/7nceVjCp//lZD/D+gaUM8GtL3T
25zbLMywm8cbxuCUuHcQjh6gVSOeMvp85qi1ozZig2TrQEgkgVsOnuWOQnIgC3pP
F+rF8IT/lcuLDiX/TfXJc7/gU7mpupmqnEfACMwOBbJmoKeWUYs+AgyqFudzWZSl
104tg5+R0/YjxBxstOs8n8ZnLmFGRGrmjFRlQahkv312V6AltpInHXJeNpBGcmbt
l9df2953CuLAH6/JAM7oFkaJE8ZSElbKHpyDvMGuoKOAPXAfUBErF1Tis74/zyJd
GOAMQAYH5tkb/G5xvdak5RY1FJGQ1NXpskvpWKmFfPAaJdENSDsd4EC4M1S85t0E
sDIfIZDWPzxqap72wBW4GXjzS03Etu7pGtpx9YCCJg6NyKP9KG+Tcowhn5WsqA3D
ViN9z1EX9plP4l+1GZ/RlPCP0UWyE0vK72BZgmhR233taI8ec/PoZEe5o7QFU0Vt
iluVzZ4Nvareyu6lP/QykLLiAxAxjq27G+tpeJtdR4S/h/tvq998OEOT76bT5RDv
MQ7Jym9If7ry1CQ7NGroR7qjUyN5uDy/wH/HEsLr1c7hveAA5f/MgloXo7X+DITI
wCjnJM9agxRdLNcmvwcAx9mpOdnZzSAcvYGl/C8oeQfbOMbwVQe4e9DbvYh1Mavq
kLbk+pCGDWsthpec0nt9XLcVilI4+0nh1VWDAyp7QWJhP4jsdfjBohsJW6Ic+I8E
c64jadInETYSH4W8cALQU/TQUftJzk45kcxVIyrHy7IoFFwHcTEW4Q==
-----END RSA PRIVATE KEY-----
```

2. Data encryption using DES algorithm and generated key

```
[nikita@Air ~/5 курс/криптография/11 Openssl2$ cat file
Data
[nikita@Air ~/5 курс/криптография/11 Openssl2$ time openssl
OpenSSL> enc -e -des -in ./file -out ./enc_file -pass file:./priv.pem -nosalt -pbkdf2
```

```
OpenSSL> q
openssl  0,01s user 0,00s system 0% cpu 1:36,56 total
[nikita@Air ~/5 курс/криптография/11 Openssl2$ cat enc_file
??K??j?%
```

3. Decryption of data using correct and incorrect keys

```
[nikita@Air ~/5 курс/криптография/11 Openssl2$ touch wrong_priv.pem
[nikita@Air ~/5 курс/криптография/11 Openssl2$ nano wrong_priv.pem
[nikita@Air ~/5 курс/криптография/11 Openssl2$ cat wrong_priv.pem
Lovushka
[nikita@Air ~/5 курс/криптография/11 Openssl2$ cat priv.pem
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-CBC,CDDA5990857F4C50

07FRag4k89+x7/SpBnCEwx5d9WQF6YTFvg5W5/7nceVjCp//lZD/D+gaUM8GtL3T
25zbLMywm8cbxuCUuHcQjh6gVSOeMvp85qi1ozZig2TrQEgkgVsOnuWOQnIgC3pP
F+rF8IT/lcuLDiX/TfXJc7/gU7mpupmqnEfACMwOBbJmoKeWUYs+AgyqFudzWZSl
104tg5+R0/YjxBxstOs8n8ZnLmFGRGrmjFRlQahkv312V6AltpInHXJeNpBGcmbt
l9df2953CuLAH6/JAM7oFkaJE8ZSElbKHpyDvMGuoKOAPXAfUBErF1Tis74/zyJd
GOAMQAYH5tkb/G5xvdak5RY1FJGQ1NXpskvpWKmFfPAaJdENSDsd4EC4M1S85t0E
sDIfIZDWPzxqap72wBW4GXjzS03Etu7pGtpx9YCCJg6NyKP9KG+Tcowhn5WsqA3D
ViN9z1EX9plP4l+1GZ/RlPCP0UWyE0vK72BZgmhR233taI8ec/PoZEe5o7QFU0Vt
iluVzZ4Nvareyu6lP/QykLLiAxAxjq27G+tpeJtdR4S/h/tvq998OEOT76bT5RDv
MQ7Jym9If7rylCQ7NGroR7qjUyN5uDy/wH/HEsLr1c7hveAA5f/MgloXo7X+DITI
wCjnJM9agxRdLNcmvwcAx9mpOdnZzSAcvYGl/C8oeQfbOMbwVQe4e9DbvYh1Mavq
kLbk+pCGDWsthpec0nt9XLcVilI4+0nh1VWDAyp7QWJhP4jsdfjBohsJW6Ic+I8E
c64jadInETYSH4W8cALQU/TQUftJzk45kcxVIyrHy7IoFFwHcTEW4Q==
-----END RSA PRIVATE KEY-----
```

```
OpenSSL> enc -d -des -in ./enc_file -out ./new_file -pass file:./wrong_priv.pem -nosalt -pbkdf2
bad decrypt
8680979072:error:06FFF064:digital envelope routines:CRYPTO_internal:bad decrypt:/AppleInternal/Library/BuildRoots/aaefc
5c95-11ed-8734-2e32217d8374/Library/Caches/com.apple.xbs/Sources/libressl/libressl-3.3/crypto/evp/evp_enc.c:549:
error in enc
```

```
OpenSSL> enc -d -des -in ./enc_file -out ./new_file -pass file:./priv.pem -nosalt -pbkd
OpenSSL> q
[nikita@Air ~/5 курс/криптография/11 Openssl2$ cat new_file
Data
```

4. Encryption and decryption of data using triple des and des ede algorithms

```
[nikita@Air ~/5 курс/криптография/11 Openssl2/3des_des_ede$ cat file
Data
[nikita@Air ~/5 курс/криптография/11 Openssl2/3des_des_ede$ openssl
OpenSSL> enc -e -des-ede3 -in ./file -out ./enc_des_ede3_file -pass file:priv.pem -nosalt -pbkdf2
OpenSSL> q
[nikita@Air ~/5 курс/криптография/11 Openssl2/3des_des_ede$ cat enc_des_ede3_file
^???vp?%
```

Decryption:

```
OpenSSL> enc -d -des-ede3 -in ./enc_des_ede3_file -out ./dec_file -pass file:priv.pem -nosalt -pbkdf2
OpenSSL> q
[nikita@Air ~/5 курс/криптография/11 Openssl2/3des_des_ede$ cat dec_file
Data
```

5. Encryption and decryption of data with RSA algorithm

```
[nikita@Air ~/5 курс/криптография/11 Openssl2$ openssl
OpenSSL> rsa -in ./priv.pem -out ./pub.pem -pubout
[Enter pass phrase for ./priv.pem:
writing RSA key
OpenSSL> q
[nikita@Air ~/5 курс/криптография/11 Openssl2$ cat pub.pem
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDAEY0InxYCbGZzXTLE3nP4q9DA
e+ob6sRbbymFAFjGBgCSRaw/+zybAzcFE3qZzkNFAEzLiIS7Iy3jrU1PPZDQcCcM
yqteRBgSFnqnf61Rpm56ZM9Ucb2dym4UAkkDXIXCRqi5Vnr2ldBifwlYcmblauik
v7KDWtr8zvDtEN5bvQIDAQAB
-----END PUBLIC KEY-----
```

Decryption:

```
[nikita@Air ~/5 курс/криптография/11 Openssl2/rsa$ openssl
OpenSSL> rsautl -encrypt -in ./file -out ./enc_file -inkey ./pub.pem -pubin
[nikita@Air ~/5 курс/криптография/11 Openssl2/rsa$ cat enc_file
A?w]Z?^*uL?4?4?hX???+<s?jn?]?r?¡?]Bn2?#????`??i???hA)?????>?q?g?tY?m???H?#u???H???|?R?I??;??y }?
[
nikita@Air ~/5 курс/криптография/11 Openssl2/rsa$ openssl
OpenSSL> rsautl -decrypt -in ./enc_file -out ./dec_file -inkey ./priv.pem
[Enter pass phrase for ./priv.pem:
OpenSSL> q
[nikita@Air ~/5 курс/криптография/11 Openssl2/rsa$ cat dec_file
Data
```

6. Encryption and decryption of data with blowfish

```
[nikita@Air ~/5 курс/криптография/11 Openssl2/bf$ openssl
OpenSSL> enc -e -bf -in ./file -out ./enc_file -pass file:priv.pem -pbkdf2
OpenSSL> q
[nikita@Air ~/5 курс/криптография/11 Openssl2/bf$ cat file
Data
[nikita@Air ~/5 курс/криптография/11 Openssl2/bf$ cat enc_file
[Salted__?\fm??x0b?;,e?%
OpenSSL> enc -d -bf -in ./enc_file -out ./dec_file -pass file:./priv.pem -pbkdf2
OpenSSL> q
[nikita@Air ~/5 курс/криптография/11 Openssl2/bf$ cat dec_file
Data
```

7. Compare of encryption time

```
[nikita@Air ~/5 курс/криптография/11 Openssl2$ openssl speed des-cbc
Doing des cbc for 3s on 16 size blocks: 16316398 des cbc's in 3.00s
Doing des cbc for 3s on 64 size blocks: 4180436 des cbc's in 3.00s
Doing des cbc for 3s on 256 size blocks: 1048570 des cbc's in 2.99s
Doing des cbc for 3s on 1024 size blocks: 262743 des cbc's in 3.00s
Doing des cbc for 3s on 8192 size blocks: 32845 des cbc's in 3.00s
LibreSSL 3.3.6
built on: date not available
options:bn(64,64) rc4(ptr,int) des(idx,cisc,16,int) aes(partial) blowfish(idx)
compiler: information not available
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
des cbc          87063.19k    89241.24k    89638.55k    89737.53k    89753.25k
```

```
[nikita@Air ~/5 курс/криптография/11 Openssl2$ openssl speed des-ede3
Doing des ede3 for 3s on 16 size blocks: 6131216 des ede3's in 2.98s
Doing des ede3 for 3s on 64 size blocks: 1564363 des ede3's in 2.99s
Doing des ede3 for 3s on 256 size blocks: 392899 des ede3's in 3.00s
Doing des ede3 for 3s on 1024 size blocks: 98103 des ede3's in 2.99s
Doing des ede3 for 3s on 8192 size blocks: 12225 des ede3's in 2.99s
LibreSSL 3.3.6
built on: date not available
options:bn(64,64) rc4(ptr,int) des(idx,cisc,16,int) aes(partial) blowfish(idx)
compiler: information not available
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
des ede3         32898.17k    33460.70k    33566.00k    33547.52k    33502.05k
```

```
[nikita@Air ~/5 курс/криптография/11 Openssl2$ openssl speed rsa
Doing 512 bit private rsa's for 10s: 176617 512 bit private RSA's in 9.98s
Doing 512 bit public rsa's for 10s: 1504768 512 bit public RSA's in 9.97s
Doing 1024 bit private rsa's for 10s: 38837 1024 bit private RSA's in 9.98s
Doing 1024 bit public rsa's for 10s: 466004 1024 bit public RSA's in 9.98s
Doing 2048 bit private rsa's for 10s: 6182 2048 bit private RSA's in 9.97s
Doing 2048 bit public rsa's for 10s: 135776 2048 bit public RSA's in 9.95s
Doing 4096 bit private rsa's for 10s: 956 4096 bit private RSA's in 9.98s
Doing 4096 bit public rsa's for 10s: 38654 4096 bit public RSA's in 9.98s
LibreSSL 3.3.6
built on: date not available
options:bn(64,64) rc4(ptr,int) des(idx,cisc,16,int) aes(partial) blowfish(idx)
compiler: information not available
                  sign    verify    sign/s verify/s
rsa  512 bits 0.000057s 0.000007s  17691.4 150962.0
rsa 1024 bits 0.000257s 0.000021s   3891.8  46701.7
rsa 2048 bits 0.001613s 0.000073s    620.0  13646.6
rsa 4096 bits 0.010444s 0.000258s     95.7   3872.8
```

The examples above show that:

• The asymmetric type of encryption presented here by the RSA algorithm is slower than the symmetric DES encryption algorithm.

• The 3DES-EDE (Encrypt-Decrypt-Encrypt) symmetric encryption algorithm shows lower performance results compared to DES-CBC

Using hashing algorithms from the openssl library.

1. PCalculation of md5 and sha1 hash values for different data

```
[nikita@Air ~/5 курс/криптография/12$ touch file
[nikita@Air ~/5 курс/криптография/12$ openssl dgst -md5 file > digests
[nikita@Air ~/5 курс/криптография/12$ openssl dgst -sha1 file >> digests
[nikita@Air ~/5 курс/криптография/12$ cat digests
MD5(file)= d41d8cd98f00b204e9800998ecf8427e
SHA1(file)= da39a3ee5e6b4b0d3255bfef95601890afd80709
[nikita@Air ~/5 курс/криптография/12$ openssl rand -hex 1000 > file
[nikita@Air ~/5 курс/криптография/12$ openssl dgst -md5 file > digests
[nikita@Air ~/5 курс/криптография/12$ openssl dgst -sha1 file >> digests
[nikita@Air ~/5 курс/криптография/12$ cat digests
MD5(file)= 4dca07d58e68efc8ae1b563189c07ff2
SHA1(file)= c86c44bc53e694f0925abbfcfc6424480db563fa
nikita@Air ~/5 курс/криптография/12$
```

2. Comparison of the running time of the md5 and sha1 hash functions

```
[nikita@Air ~/5 курс/криптография/12$ openssl speed md5
Doing md5 for 3s on 16 size blocks: 20145602 md5's in 2.99s
Doing md5 for 3s on 64 size blocks: 12101023 md5's in 2.99s
Doing md5 for 3s on 256 size blocks: 5276360 md5's in 2.99s
Doing md5 for 3s on 1024 size blocks: 1612564 md5's in 2.98s
Doing md5 for 3s on 8192 size blocks: 217765 md5's in 2.99s
LibreSSL 3.3.6
built on: date not available
options:bn(64,64) rc4(ptr,int) des(idx,cisc,16,int) aes(partial) blowfish(idx)
compiler: information not available
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
md5            107644.58k   259095.24k   452311.17k   554994.07k   595804.45k
[nikita@Air ~/5 курс/криптография/12$ openssl speed sha1
Doing sha1 for 3s on 16 size blocks: 24110775 sha1's in 3.00s
Doing sha1 for 3s on 64 size blocks: 14796204 sha1's in 3.00s
Doing sha1 for 3s on 256 size blocks: 6447437 sha1's in 2.99s
Doing sha1 for 3s on 1024 size blocks: 1986350 sha1's in 2.99s
Doing sha1 for 3s on 8192 size blocks: 265249 sha1's in 2.99s
LibreSSL 3.3.6
built on: date not available
options:bn(64,64) rc4(ptr,int) des(idx,cisc,16,int) aes(partial) blowfish(idx)
compiler: information not available
The 'numbers' are in 1000s of bytes per second processed.
type             16 bytes     64 bytes    256 bytes   1024 bytes   8192 bytes
sha1           128735.76k   316102.69k   551809.31k   679508.27k   726376.78k
```

3. Get a File Authenticator Using the DES-CBC Algorithm

```
[nikita@Air ~/5 курс/криптография/12$ openssl rand -hex 10000 > file
[nikita@Air ~/5 курс/криптография/12$ openssl
OpenSSL> enc -e -des-cbc -in ./file -out ./enc_file_1 -pass pass:1 -nosalt -pbkdf2
OpenSSL> enc -e -des-cbc -in ./file -out ./enc_file_2 -pass pass:1 -nosalt -pbkdf2
OpenSSL> q
[nikita@Air ~/5 курс/криптография/12$ diff enc_file_1 enc_file_2
[nikita@Air ~/5 курс/криптография/12$ openssl dgst -sha1 enc_file_1
SHA1(enc_file_1)= 07a5170fa6008ba668fbcbbd83e4068ab00df99f
[nikita@Air ~/5 курс/криптография/12$ openssl dgst -sha1 enc_file_2
SHA1(enc_file_2)= 07a5170fa6008ba668fbcbbd83e4068ab00df99f
```

4. Performing a hash function using keys

4.1. Create private key of RSA without password

```
OpenSSL> genpkey -algorithm rsa -out priv.rsa
.....................................................................
.....................................................................
.....................................................+++++
.+++++
OpenSSL> rsa -in ./priv.rsa -text -noout
RSA Private-Key: (2048 bit)
modulus:
    00:d0:0e:d6:b4:c2:ee:1a:e8:1c:43:ba:d1:18:18:
    75:54:97:94:8f:41:ca:99:4c:f5:15:04:ed:db:b1:
    32:89:32:d1:e3:1c:fc:39:1e:7f:4d:a3:d0:e9:58:
    4b:d2:f4:e5:82:9e:9e:aa:ac:0e:cf:a1:a1:d3:fc:
    c4:3b:5d:85:5f:09:72:37:ce:24:01:89:6b:78:ef:
    64:4d:81:68:da:89:7c:0b:36:dd:38:fa:f0:f7:79:
    82:cc:8d:30:11:e3:4e:92:26:2f:7e:88:b3:18:47:
    65:79:dc:bf:2e:fc:ef:8c:28:6b:f6:35:2d:4a:26:
    75:8a:69:3e:72:6e:95:b5:93:88:05:17:70:0f:ab:
    41:60:be:69:d4:2c:33:cb:39:ad:6b:c3:57:82:94:
    88:88:77:f6:1a:e7:d2:2d:4a:5f:04:d9:2b:0f:6c:
    fd:27:3c:e7:3a:af:65:de:6e:ac:61:84:0c:d3:fc:
    0f:95:b7:db:e1:b5:38:22:3a:a6:bb:b2:d8:98:58:
    b6:12:9d:ae:b4:77:d0:bb:3d:d7:2d:d1:d9:34:a6:
    29:4b:a2:41:15:14:f6:6e:7c:17:cb:5a:52:31:85:
    f4:e4:62:ff:6d:85:92:1e:8a:09:18:06:c6:e2:67:
    4f:fa:c2:1f:ea:e5:8f:fe:5a:dd:b9:fc:f1:3c:b0:
    1f:89
publicExponent: 65537 (0x10001)
privateExponent:
    10:0d:e5:11:63:ad:3d:d4:45:42:10:ab:4b:c1:af:
    64:0c:a2:40:ff:a1:a3:7c:a5:b8:ae:7d:b0:23:17:
    34:31:00:b3:16:ac:7b:b7:d7:b8:e4:f8:1c:d9:5c:
    58:75:df:33:da:0b:82:3d:ee:92:a4:f5:38:c7:5b:
    58:fd:59:6a:40:ef:58:51:06:c4:3b:97:58:68:98:
    83:c6:85:91:bf:64:1b:f5:6a:d2:97:c2:7a:46:1d:
    0a:ad:a6:54:eb:06:48:0e:bc:41:76:48:e3:89:b6:
    d1:d5:6f:c7:2f:6b:48:94:61:e8:48:a8:2a:fd:96:
    ba:4c:6e:ae:23:cc:57:8b:5c:25:cc:ed:ec:a7:67:
    11:9c:f5:27:f0:5e:77:34:b3:05:7f:2e:9d:62:d6:
    15:32:fc:5f:d4:5d:93:a0:e2:48:95:f2:42:82:c5:
    ff:ff:0f:42:75:38:d8:53:23:08:ec:7e:13:94:5a:
    0f:a5:84:f0:56:ff:78:2c:d7:67:c4:52:e6:b6:b6:
    41:85:80:6a:28:bf:08:67:82:65:99:41:12:7f:89:
    10:d0:83:ff:bc:d1:36:4c:64:29:04:79:f9:56:39:
    75:8b:8c:c7:30:f5:96:88:2b:0a:cd:d4:4b:46:5d:
    1f:ee:64:b7:fe:ec:a2:ec:95:32:55:99:68:56:5a:
    95
prime1:
    00:ec:a3:41:e2:1f:92:2a:3a:75:e1:37:0a:0c:83:
    42:b4:3f:ca:34:77:c6:8b:3b:31:d9:04:61:eb:23:
    4b:cc:35:a9:45:62:46:bd:fa:a8:08:e7:2e:d9:af:
    82:1a:9c:75:5f:8d:56:b2:f4:37:21:20:7b:3b:2b:
    af:36:d8:d2:9f:74:0c:b4:12:b5:05:c8:29:aa:80:
    36:dd:db:8b:40:e8:8d:84:94:2a:d0:db:c7:cc:1b:
    d7:6a:8c:c5:a3:d9:f8:b1:44:fb:aa:78:f5:ce:9e:
    41:f7:37:a8:f8:24:0f:42:64:f5:c0:fd:fc:79:81:
    e5:6b:ca:5d:ea:ed:cd:4a:07
```

4.2. RSA public key generation with relative public key

```
[nikita@Air ~/5 курс/криптография/12/rsa$ openssl
OpenSSL> rsa -in ./priv.rsa -out ./pub.rsa -pubout -outform PEM
writing RSA key
OpenSSL> q
[nikita@Air ~/5 курс/криптография/12/rsa$ cat pub.rsa
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0A7WtMLuGugcQ7rRGBh1
VJeUj0HKmUz1FQTt27EyiTLR4xz8OR5/TaPQ6VhL0vTlgp6eqqwOz6Gh0/zEO12F
XwlyN84kAYlreO9kTYFo2ol8CzbdOPrw93mCzI0wEeNOkiYvfoizGEdledy/Lvzv
jChr9jUtSiZ1imk+cm6VtZOIBRdwD6tBYL5p1Cwzyzmta8NXgpSIiHf2GufSLUpf
BNkrD2z9JzznOq9l3m6sYYQM0/wPlbfb4bU4Ijqmu7LYmFi2Ep2utHfQuz3XLdHZ
NKYpS6JBFRT2bnwXy1pSMYX05GL/bYWSHooJGAbG4mdP+sIf6uWP/lrdufzxPLAf
iQIDAQAB
-----END PUBLIC KEY-----
```

4.3. Creating and verifying signature of file

```
[nikita@Air ~/5 курс/криптография/12/rsa$ openssl rand -hex 1024 > file
[nikita@Air ~/5 курс/криптография/12/rsa$ openssl
OpenSSL> dgst -sha256 -sign priv.rsa -out file.sign file
OpenSSL> q
[nikita@Air ~/5 курс/криптография/12/rsa$ cp file.sign modified.sign
[nikita@Air ~/5 курс/криптография/12/rsa$ nano modified.sign
[nikita@Air ~/5 курс/криптография/12/rsa$ xxd file.sign | head -n 2
00000000: c9ec fe24 d323 f433 d3fa bad0 cea8 e844  ...$.#.3.......D
00000010: 1fbc 6753 b488 17e2 ed16 4f87 4904 288f  ..gS......O.I.(.
[nikita@Air ~/5 курс/криптография/12/rsa$ xxd modified.sign | head -n 2
00000000: 3f3f 243f 3f3f 3f3f 3f3f 1f3f 6753 3f3f  ??$???????.?gS??
00000010: 173f 164f 3f49 0428 3f3f 1b3f 3f57 52ec  .?.O?I.(??.??WR.
```

```
OpenSSL> dgst -sha256 file
SHA256(file)= abf77d90b2e529f2b7de5cb054f921e12737ca05288e484c587cb0ee51cfb1aa
OpenSSL> dgst -sha256 file.sign
SHA256(file.sign)= 3487946eb01ac35d251340f7a86c5b1a583d39be9519380821ad230835aed3dd
OpenSSL> dgst -sha256 modified.sign
SHA256(modified.sign)= f0aadb043b5144cff6b83c004e7080105be303987023da7f5429a26cbbbb6981
OpenSSL> dgst -sha512 -verify pub.rsa -signature file.sign file
Verification Failure
error in dgst
```

5. Working with certificates

5.1. Creating a self-signed x509 certificate

```
[nikita@Air ~/5 курс/криптография/12/4.cert$ openssl
OpenSSL> req -new -x509 -days 365 -nodes -out cet.pem -keyout priv_cert_key.key
Generating a 2048 bit RSA private key
..................+++++
.............+++++
writing new private key to 'priv_cert_key.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:cat cert.pem | head -n 4
string is too long, it needs to be less than  2 bytes long
Country Name (2 letter code) []:Ni
State or Province Name (full name) []:Rostov region
Locality Name (eg, city) []:Rostov-on-Don
Organization Name (eg, company) []:VKB
Organizational Unit Name (eg, section) []:Nikita
Common Name (eg, fully qualified host name) []:pochta@mail.ru
Email Address []:pochta@mail.ru
OpenSSL> q
[nikita@Air ~/5 курс/криптография/12/4.cert$ cat cert.pem | head -n 4
cat: cert.pem: No such file or directory
[nikita@Air ~/5 курс/криптография/12/4.cert$ ls
cet.pem           priv_cert_key.key
[nikita@Air ~/5 курс/криптография/12/4.cert$ cat cet.pem | head -n 4
-----BEGIN CERTIFICATE-----
MIIDpjCCAo4CCQCiA5Vpa9mtMTANBgkqhkiG9w0BAQsFADCBlDELMAkGA1UEBhMC
TmkxFjAUBgNVBAgMDVJvc3RvdiByZWdpb24xFjAUBgNVBAcMDVJvc3Rvdi1vbi1E
b24xDDAKBgNVBAoMA1ZLQjEPMA0GA1UECwwGTmlraXRhMRcwFQYDVQQDDA5wb2No
[nikita@Air ~/5 курс/криптография/12/4.cert$ cat priv_cert_key.key | head -n 3
-----BEGIN PRIVATE KEY-----
MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQC2lKvFCHfLaxDB
0Hlc0NPwW3V54j1zNj0/CiVC5B5Rg9IcRhP49fqPk+fzotkHe6Vjgd5umfsdgw0N
```

5.2. Checking the certificate and displaying information about it

```
nikita@Air ~/5 курс/криптография/12/4.cert$ openssl
OpenSSL> verify ./cet.pem
C = Ni, ST = Rostov region, L = Rostov-on-Don, O = VKB, OU = Nikita, CN = pochta@mail.ru,
 emailAddress = pochta@mail.ru
error 18 at 0 depth lookup:self signed certificate
./cet.pem: verification failed: 18 (self signed certificate)
error in verify
```

```
[nikita@Air ~/5 курс/криптография/12/4.cert$ openssl x509 -in ./cet.pem -text -noout                ]
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number:
            a2:03:95:69:6b:d9:ad:31
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=Ni, ST=Rostov region, L=Rostov-on-Don, O=VKB, OU=Nikita, CN=pochta@mail.ru/emailAd
dress=pochta@mail.ru
        Validity
            Not Before: Feb 20 12:51:35 2023 GMT
            Not After : Feb 20 12:51:35 2024 GMT
        Subject: C=Ni, ST=Rostov region, L=Rostov-on-Don, O=VKB, OU=Nikita, CN=pochta@mail.ru/emailA
ddress=pochta@mail.ru
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:b6:94:ab:c5:08:77:cb:6b:10:c1:d0:79:5c:d0:
                    d3:f0:5b:75:79:e2:3d:73:36:3d:3f:0a:25:42:e4:
                    1e:51:83:d2:1c:46:13:f8:f5:fa:8f:93:e7:f3:a2:
                    d9:07:7b:a5:63:81:de:6e:99:fb:1d:83:0d:0d:83:
                    bb:31:ad:91:d1:73:d5:41:99:10:f1:4e:91:6a:aa:
                    75:fb:77:6a:0c:dc:20:bb:25:b2:c9:6e:7e:49:a6:
                    39:2f:e7:e4:6f:ad:d7:8f:72:e7:53:c1:28:79:37:
                    7b:e2:db:06:27:c1:98:18:30:53:f6:d9:3c:44:c0:
                    b4:7e:1a:51:b5:f2:a1:f1:ae:33:68:77:f4:5a:8b:
                    26:0d:5a:99:d5:7f:08:2f:db:4c:78:ca:26:37:d1:
                    e4:82:5e:78:1c:41:be:d7:5d:2e:67:ff:1f:5a:15:
                    7b:5b:23:0b:fa:b6:67:5e:92:ee:3f:ca:24:60:33:
                    2c:a4:a7:64:b6:52:ba:f2:d1:4a:17:bf:0f:9c:c8:
                    c2:64:dd:cd:3d:71:e0:96:6c:82:1a:ab:72:18:4e:
                    c0:a4:fe:36:94:f4:7e:17:94:21:59:08:b0:80:51:
                    3f:04:09:bf:a0:11:92:d2:5b:75:73:19:de:c9:7b:
                    88:21:02:8d:cb:aa:c9:c6:17:95:bb:ea:41:47:88:
                    ca:4d
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha256WithRSAEncryption
         12:c3:6c:fe:60:c9:ce:4a:3c:cc:18:d0:7f:4b:af:4a:a2:56:
         a4:b0:d6:d0:c6:47:12:8b:1e:94:c5:58:d1:e4:ac:9d:40:5c:
         00:17:bc:37:a4:71:13:05:4b:45:cb:1a:64:c3:6b:ef:95:2f:
         79:1c:9c:28:a3:91:7a:87:fe:81:8e:3a:d1:e4:c2:8f:99:2e:
         ff:15:03:2a:72:83:e1:f8:d9:eb:cc:70:94:69:5a:68:79:40:
         88:99:b8:a5:1a:46:4b:6c:fe:2f:07:d0:05:85:de:eb:57:ba:
         16:56:3e:3c:79:a7:dd:0c:be:1c:5b:76:a9:bd:25:8c:2d:58:
         25:76:c2:2c:59:63:77:1e:2f:9b:76:bc:5c:7e:8b:91:b7:3f:
         7b:e0:05:92:06:2b:d1:d2:ac:ec:3b:6f:70:70:1c:2f:46:80:
         61:4b:1c:93:0c:10:f6:36:0e:a2:ed:b7:21:bc:c0:50:bc:1d:
         c5:81:d0:09:01:15:d1:b1:48:a2:fc:df:ca:d1:ff:a1:f9:dd:
         91:28:8a:c2:e8:fa:6b:5b:5a:65:65:a9:a9:18:cf:59:d1:31:
         80:52:63:83:95:0a:62:dc:bb:0f:9e:0d:68:a1:f7:ae:1b:6e:
         3c:52:9f:91:21:62:b8:16:0f:90:51:02:72:de:1a:e0:a6:86:
         63:44:44:c6
```

5.3. Signature verification using the public key contained in the certificate

```
[nikita@Air ~/5 курс/криптография/12/4.cert$ openssl rand -hex 1024 > file
[nikita@Air ~/5 курс/криптография/12/4.cert$ openssl
OpenSSL> dgst -sha256 -sign priv_cert_key.key -out file.sign file
OpenSSL> q
[nikita@Air ~/5 курс/криптография/12/4.cert$ xxd file.sign | head -n 2
00000000: 0ded fd83 2d03 17d1 1a82 6d53 1701 8db4  ....-.....mS....
00000010: 636d 5b79 9da8 e189 2a4b 479f 625a f56b  cm[y....*KG.bZ.k
[nikita@Air ~/5 курс/криптография/12/4.cert$ openssl x509 -in cet.pem -noout -out pub_cert_key.key -p
ubkey > pub_cert_key.key
[nikita@Air ~/5 курс/криптография/12/4.cert$ cat pub_cert_key.key
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtpSrxQh3y2sQwdB5XNDT
8Ft1eeI9czY9PwolQuQeUYPSHEYT+PX6j5Pn86LZB3ulY4Hebpn7HYMNDYO7Ma2R
0XPVQZkQ8U6Raqp1+3dqDNwguyWyyW5+SaY5L+fkb63Xj3LnU8EoeTd74tsGJ8GY
GDBT9tk8RMC0fhpRtfKh8a4zaHf0WosmDVqZ1X8IL9tMeMomN9Hkgl54HEG+110u
Z/8fWhV7WyML+rZnXpLuP8okYDMspKdktlK68tFKF78PnMjCZN3NPXHglmyCGqty
GE7ApP421lPR+F5QhWQiwgFE/BAm/oBGS0lt1cxneyXuIIQKNy6rJxheVu+pBR4jK
TQIDAQAB
-----END PUBLIC KEY-----
[nikita@Air ~/5 курс/криптография/12/4.cert$ openssl
OpenSSL> dgst -sha256 -verify pub_cert_key.key -signature file.sign file
Verified OK
OpenSSL> dgst -sha256 -verify pub_cert_key.key -signature modified.sign file
Verification Failure
error in dgst
```