1-3 学时实践要求（30 分）

1. 参考云班课相关教学视频，在 Ubuntu或openEuler中（推荐 openEuler）中实践课程思维导图中OpenSSL相关内容，使用Markdown记录详细记录实践过程，每完成一项git commit 一次。（5分）

- 实践过程：
  - openssl version

```
root@Youer:~# openssl version
OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)
```

  - openssl list -help

```
root@Youer:~# openssl list -help
Usage: list [options]

General options:
-help                      Display this summary

Output options:
-1                         List in one column
-verbose                   Verbose listing
-select val                Select a single algorithm
-commands                  List of standard commands
-standard-commands         List of standard commands
-digest-commands           List of message digest commands (deprecated)
-digest-algorithms         List of message digest algorithms
-kdf-algorithms            List of key derivation and pseudo random
function algorithms
-random-instances          List the primary, public and private random
number generator details
-random-generators         List of random number generators
-mac-algorithms            List of message authentication code
algorithms
-cipher-commands           List of cipher commands (deprecated)
-cipher-algorithms         List of cipher algorithms
-encoders                  List of encoding methods
-decoders                  List of decoding methods
-key-managers              List of key managers
-key-exchange-algorithms   List of key exchange algorithms
-kem-algorithms            List of key encapsulation mechanism
algorithms
-signature-algorithms      List of signature algorithms
-asymcipher-algorithms     List of asymmetric cipher algorithms
-public-key-algorithms     List of public key algorithms
-public-key-methods        List of public key methods
-store-loaders             List of store loaders
-providers                 List of provider information
-engines                   List of loaded engines
```

```
-disabled                 List of disabled features
-options val              List options for specified command
-objects                  List built in objects (OID<->name mappings)

Provider options:
-provider-path val        Provider load path (must be before 'provider'
argument if required)
-provider val             Provider to load (can be specified multiple
times)
-propquery val            Property query used when fetching algorithms
```

- openssl -help

```
root@Youer:~#  openssl -help
help:

Standard commands
asn1parse         ca                ciphers           cmp
cms               crl               crl2pkcs7         dgst
dhparam           dsa               dsaparam          ec
ecparam           enc               engine            errstr
fipsinstall       gendsa            genpkey           genrsa
help              info              kdf               list
mac               nseq              ocsp              passwd
pkcs12            pkcs7             pkcs8             pkey
pkeyparam         pkeyutl           prime             rand
rehash            req               rsa               rsautl
s_client          s_server          s_time            sess_id
smime             speed             spkac             srp
storeutl          ts                verify            version
x509

Message Digest commands (see the `dgst' command for more details)
blake2b512        blake2s256        md4               md5
rmd160            sha1              sha224            sha256
sha3-224          sha3-256          sha3-384          sha3-512
sha384            sha512            sha512-224        sha512-256
shake128          shake256          sm3

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       aria-128-cbc      aria-128-cfb
aria-128-cfb1     aria-128-cfb8     aria-128-ctr      aria-128-ecb
aria-128-ofb      aria-192-cbc      aria-192-cfb      aria-192-cfb1
aria-192-cfb8     aria-192-ctr      aria-192-ecb      aria-192-ofb
aria-256-cbc      aria-256-cfb      aria-256-cfb1     aria-256-cfb8
aria-256-ctr      aria-256-ecb      aria-256-ofb      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         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             seed               seed-cbc           seed-cfb
seed-ecb           seed-ofb           sm4-cbc            sm4-cfb
sm4-ctr            sm4-ecb            sm4-ofb
```

- 数据输入与输出
    - 文本

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo 123 | openssl sm3
SM3(stdin)= e95001aed4b6f7de59169913997dace404f05091ed49c37133a9950a69405a9c
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo "123" | openssl sm3
SM3(stdin)= e95001aed4b6f7de59169913997dace404f05091ed49c37133a9950a69405a9c
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo 123 | od -tx1 -tc
0000000  31  32  33  0a
        1   2   3  \n
0000004
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo -n 123 | od -tx1 -tc
0000000  31  32  33
        1   2   3
0000003
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo 123 | openssl sm3
SM3(stdin)= e95001aed4b6f7de59169913997dace404f05091ed49c37133a9950a69405a9c
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo -n 123 | openssl sm3
SM3(stdin)= 6e0f9e14344c5406a0cf5a3b4dfb665f87f4a771a31f7edbb5c72874a32b2957
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo 123 > 123.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl sm3 -file 123.txt
SM3(123.txt)= e95001aed4b6f7de59169913997dace404f05091ed49c37133a9950a69405a9c
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo 123 | openssl sm3
SM3(stdin)= e95001aed4b6f7de59169913997dace404f05091ed49c37133a9950a69405a9c
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git commit -m
```

```
"Text Data Input and Output"
[master (root-commit) 4ff1a00] Text Data Input and Output
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are
accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to
edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit
with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 shiyan1-1/openssl/123.txt
```

- 二进制（16进制）

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo
"obase=16;123" | bc
7B
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo -n -e "\x7B"
> 123.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# od -tx1 123.bin
0000000 7b
0000001
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl sm3 -file
123.bin
SM3(123.bin)=
2ed59fea0dbe4e4f02de67ee657eb6be8e22a7db425103402d8a36d7b6f6d344
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo -ne "\x7B" |
openssl sm3
SM3(stdin)=
2ed59fea0dbe4e4f02de67ee657eb6be8e22a7db425103402d8a36d7b6f6d344
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# ls
123.bin  123.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git commit -m
"Input and Output of Data in Different Bases"
[master 3dce8b6] Input and Output of Data in Different Bases
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are
accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to
edit
```

```
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 shiyan1-1/openssl/123.bin
```

- 常用命令
  - prime

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl prime -help
Usage: prime [options] [number...]

General options:
-help                 Display this summary
-bits +int            Size of number in bits
-checks +int          Number of checks

Output options:
-hex                  Hex output
-generate             Generate a prime
-safe                 When used with -generate, generate a safe prime

Provider options:
-provider-path val    Provider load path (must be before 'provider' argument if required)
-provider val         Provider to load (can be specified multiple times)
-propquery val        Property query used when fetching algorithms

Parameters:
number                Number(s) to check for primality if not generating
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl prime 3
3 (3) is prime
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl prime 33
21 (33) is not prime
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl prime -checks 10 33
21 (33) is not prime
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl prime -hex 4F
4F (4F) is prime
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl prime -
```

```
generate -bits 10
809
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl prime 809
329 (809) is prime
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl prime -generate -bits 10
947
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl prime 947
3B3 (947) is prime
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl prime -generate -bits 10 -hex
03B3
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl prime -hex 03B3
3B3 (03B3) is prime
```

- rand

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl rand -help
Usage: rand [options] num

General options:
-help                Display this summary
-engine val          Use engine, possibly a hardware device

Output options:
-out outfile         Output file
-base64              Base64 encode output
-hex                 Hex encode output

Random state options:
-rand val            Load the given file(s) into the random number generator
-writerand outfile   Write random data to the specified file

Provider options:
-provider-path val   Provider load path (must be before 'provider' argument if required)
-provider val        Provider to load (can be specified multiple times)
-propquery val       Property query used when fetching algorithms

Parameters:
num                  Number of bytes to generate
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl rand 10
���r]�&O�root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl rand 10 | od -tx1
0000000 e1 7b 3d 20 90 63 96 80 99 b4
0000012
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl rand 10 | xxd -p
5dc66a8b55353d23dbb1
```

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl rand -hex 10
399ce608f47015551a56
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl rand -base64 10
7lSOCg0mxCNr8A==
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl rand -out r1.bin 10
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# od -tx1 r1.bin
0000000 18 cc 43 eb ff ab 86 01 61 82
0000012
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl rand 10 > r2.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# cat r2.bin | xxd -p
abe13e7faa057c3f7c62
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# ls
123.bin  123.txt  r1.bin  r2.bin
```

- base64

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl base64 -help
Usage: base64 [options]

General options:
-help               Display this summary
-list               List ciphers
-ciphers            Alias for -list
-e                  Encrypt
-d                  Decrypt
-p                  Print the iv/key
-P                  Print the iv/key and exit
-engine val         Use engine, possibly a hardware device

Input options:
-in infile          Input file
-k val              Passphrase
-kfile infile       Read passphrase from file

Output options:
-out outfile        Output file
-pass val           Passphrase source
-v                  Verbose output
-a                  Base64 encode/decode, depending on encryption flag
-base64             Same as option -a
-A                  Used with -[base64|a] to specify base64 buffer as a single line

Encryption options:
-nopad              Disable standard block padding
```

```
-salt               Use salt in the KDF (default)
-nosalt             Do not use salt in the KDF
-debug              Print debug info
-bufsize val        Buffer size
-K val              Raw key, in hex
-S val              Salt, in hex
-iv val             IV in hex
-md val             Use specified digest to create a key from the
passphrase
-iter +int          Specify the iteration count and force use of
PBKDF2
-pbkdf2             Use password-based key derivation function 2
-none               Don't encrypt
-*                  Any supported cipher

Random state options:
-rand val           Load the given file(s) into the random number
generator
-writerand outfile  Write random data to the specified file

Provider options:
-provider-path val  Provider load path (must be before 'provider'
argument if required)
-provider val       Provider to load (can be specified multiple
times)
-propquery val      Property query used when fetching algorithms
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo xlm | openssl
base64
eGxtCg==
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo xlm | openssl
base64 -e
eGxtCg==
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo eGxtCg== |
openssl base64 -d
xlm
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo -ne
"\x11\x22\x33" | openssl base64
ESIz
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo ESIz |
openssl base64 -d | xxd -p
112233
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo -ne
"\x11\x22\x33\x44" | openssl base64
ESIzRA==
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo ESIzRA== |
openssl base64 -d | xxd -p
11223344
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo xlm > xlm.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl base64 -in
xlm.txt -out xlm.b64
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# cat xlm.b64
eGxtCg==
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl base64 -d
-in xlm.b64 -out xlm2.txt
```

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# diff xlm.txt
xlm2.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# cat xlm2.txt
xlm
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git commit -m
"finish base64 command"
[master 9c2859a] finish base64 command
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are
accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to
edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit
with:

    git commit --amend --reset-author

5 files changed, 5 insertions(+)
create mode 100644 shiyan1-1/openssl/r1.bin
create mode 100644 shiyan1-1/openssl/r2.bin
create mode 100644 shiyan1-1/openssl/xlm.b64
create mode 100644 shiyan1-1/openssl/xlm.txt
create mode 100644 shiyan1-1/openssl/xlm2.txt
```

- asn1parse

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl asn1parse
-help
Usage: asn1parse [options]

General options:
-help             Display this summary
-oid infile       file of extra oid definitions

I/O options:
-inform PEM|DER   input format - one of DER PEM
-in infile        input file
-out outfile      output file (output format is always DER)
-noout            do not produce any output
-offset +int      offset into file
-length +int      length of section in file
-strparse +int    offset; a series of these can be used to 'dig'
-genstr val       string to generate ASN1 structure from
                 into multiple ASN1 blob wrappings
-genconf val      file to generate ASN1 structure from
```

```
-strictpem       do not attempt base64 decode outside PEM markers
-item val        item to parse and print
                (-inform  will be ignored)

Formatting options:
-i               indents the output
-dump            unknown data in hex form
-dlimit +int     dump the first arg bytes of unknown data in hex
form
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo -ne
"\x03\x02\x04\x90" >bitstring.der
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl asn1parse
-inform der -i -in bitstring.der
    0:d=0  hl=2 l=   2 prim: BIT STRING
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl base64 -in
bitstring.der -out bitstring.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# ls bitstring.pem
bitstring.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl asn1parse
-inform PEM -in bitstring.pem
    0:d=0  hl=2 l=   2 prim: BIT STRING
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git commit -m
"finish asn1parse command"
[master 0f82610] finish asn1parse command
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are
accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to
edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit
with:

    git commit --amend --reset-author

2 files changed, 2 insertions(+)
create mode 100644 shiyan1-1/openssl/bitstring.der
create mode 100644 shiyan1-1/openssl/bitstring.pem
```

- dgst

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl dgst -help
Usage: dgst [options] [file...]

General options:
-help               Display this summary
```

```
-list               List digests
-engine val         Use engine e, possibly a hardware device
-engine_impl        Also use engine given by -engine for digest
operations
-passin val         Input file pass phrase source

Output options:
-c                  Print the digest with separating colons
-r                  Print the digest in coreutils format
-out outfile        Output to filename rather than stdout
-keyform format     Key file format (ENGINE, other values ignored)
-hex                Print as hex dump
-binary             Print in binary form
-xoflen +int        Output length for XOF algorithms
-d                  Print debug info
-debug              Print debug info

Signing options:
-sign val           Sign digest using private key
-verify val         Verify a signature using public key
-prverify val       Verify a signature using private key
-sigopt val         Signature parameter in n:v form
-signature infile   File with signature to verify
-hmac val           Create hashed MAC with key
-mac val            Create MAC (not necessarily HMAC)
-macopt val         MAC algorithm parameters in n:v form or key
-*                  Any supported digest
-fips-fingerprint   Compute HMAC with the key used in OpenSSL-FIPS
fingerprint

Random state options:
-rand val           Load the given file(s) into the random number
generator
-writerand outfile  Write random data to the specified file

Provider options:
-provider-path val  Provider load path (must be before 'provider'
argument if required)
-provider val       Provider to load (can be specified multiple times)
-propquery val      Property query used when fetching algorithms

Parameters:
file                Files to digest (optional; default is stdin)
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl dgst -list
Supported digests:
-blake2b512                -blake2s256                -md4
-md5                       -md5-sha1                  -ripemd
-ripemd160                 -rmd160                    -sha1
-sha224                    -sha256                    -sha3-224
-sha3-256                  -sha3-384                  -sha3-512
-sha384                    -sha512                    -sha512-224
-sha512-256                -shake128                  -shake256
-sm3                       -ssl3-md5                  -ssl3-sha1
-whirlpool
```

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo xlm | openssl dgst
-sm3
SM3(stdin)=
0d7c54df40fee120d0d41356333b22aec2556ecf3961ce539196a5b95e38c0f7
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo xlm | openssl sm3
SM3(stdin)=
0d7c54df40fee120d0d41356333b22aec2556ecf3961ce539196a5b95e38c0f7
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo xlm | openssl sm3
-hex
SM3(stdin)=
0d7c54df40fee120d0d41356333b22aec2556ecf3961ce539196a5b95e38c0f7
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo xlm | openssl sm3
-binary
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo xlm | openssl sm3
-binary | xxd -p
0d7c54df40fee120d0d41356333b22aec2556ecf3961ce539196a5b95e38
c0f7
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl#  echo xlm > xlm.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl sm3 xlm.txt
SM3(xlm.txt)=
0d7c54df40fee120d0d41356333b22aec2556ecf3961ce539196a5b95e38c0f7
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# echo xlm | openssl sm3
SM3(stdin)=
0d7c54df40fee120d0d41356333b22aec2556ecf3961ce539196a5b95e38c0f7
```

- enc

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl enc -help
Usage: enc [options]

General options:
-help                Display this summary
-list                List ciphers
-ciphers             Alias for -list
-e                   Encrypt
-d                   Decrypt
-p                   Print the iv/key
-P                   Print the iv/key and exit
-engine val          Use engine, possibly a hardware device

Input options:
-in infile           Input file
-k val               Passphrase
-kfile infile        Read passphrase from file

Output options:
-out outfile         Output file
-pass val            Passphrase source
-v                   Verbose output
-a                   Base64 encode/decode, depending on encryption flag
-base64              Same as option -a
-A                   Used with -[base64|a] to specify base64 buffer as a
```

```
single line

Encryption options:
-nopad               Disable standard block padding
-salt                Use salt in the KDF (default)
-nosalt              Do not use salt in the KDF
-debug               Print debug info
-bufsize val         Buffer size
-K val               Raw key, in hex
-S val               Salt, in hex
-iv val              IV in hex
-md val              Use specified digest to create a key from the
passphrase
-iter +int           Specify the iteration count and force use of PBKDF2
-pbkdf2              Use password-based key derivation function 2
-none                Don't encrypt
-*                   Any supported cipher

Random state options:
-rand val            Load the given file(s) into the random number
generator
-writerand outfile  Write random data to the specified file

Provider options:
-provider-path val  Provider load path (must be before 'provider'
argument if required)
-provider val        Provider to load (can be specified multiple times)
-propquery val       Property query used when fetching algorithms
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl enc -list
Supported ciphers:
-aes-128-cbc               -aes-128-cfb               -aes-128-cfb1
-aes-128-cfb8              -aes-128-ctr               -aes-128-ecb
-aes-128-ofb               -aes-192-cbc               -aes-192-cfb
-aes-192-cfb1              -aes-192-cfb8              -aes-192-ctr
-aes-192-ecb               -aes-192-ofb               -aes-256-cbc
-aes-256-cfb               -aes-256-cfb1              -aes-256-cfb8
-aes-256-ctr               -aes-256-ecb               -aes-256-ofb
-aes128                    -aes128-wrap               -aes192
-aes192-wrap               -aes256                    -aes256-wrap
-aria-128-cbc              -aria-128-cfb              -aria-128-cfb1
-aria-128-cfb8             -aria-128-ctr              -aria-128-ecb
-aria-128-ofb              -aria-192-cbc              -aria-192-cfb
-aria-192-cfb1             -aria-192-cfb8             -aria-192-ctr
-aria-192-ecb              -aria-192-ofb              -aria-256-cbc
-aria-256-cfb              -aria-256-cfb1             -aria-256-cfb8
-aria-256-ctr              -aria-256-ecb              -aria-256-ofb
-aria128                   -aria192                   -aria256
-bf                        -bf-cbc                    -bf-cfb
-bf-ecb                    -bf-ofb                    -blowfish
-camellia-128-cbc          -camellia-128-cfb          -camellia-128-
cfb1
-camellia-128-cfb8         -camellia-128-ctr          -camellia-128-ecb
-camellia-128-ofb          -camellia-192-cbc          -camellia-192-cfb
-camellia-192-cfb1         -camellia-192-cfb8         -camellia-192-ctr
```

```
-camellia-192-ecb          -camellia-192-ofb          -camellia-256-cbc
-camellia-256-cfb          -camellia-256-cfb1         -camellia-256-
cfb8
-camellia-256-ctr          -camellia-256-ecb          -camellia-256-ofb
-camellia128               -camellia192               -camellia256
-cast                      -cast-cbc                  -cast5-cbc
-cast5-cfb                 -cast5-ecb                 -cast5-ofb
-chacha20                  -des                       -des-cbc
-des-cfb                   -des-cfb1                  -des-cfb8
-des-ecb                   -des-ede                   -des-ede-cbc
-des-ede-cfb               -des-ede-ecb               -des-ede-ofb
-des-ede3                  -des-ede3-cbc              -des-ede3-cfb
-des-ede3-cfb1             -des-ede3-cfb8             -des-ede3-ecb
-des-ede3-ofb              -des-ofb                   -des3
-des3-wrap                 -desx                      -desx-cbc
-id-aes128-wrap            -id-aes128-wrap-pad        -id-aes192-wrap
-id-aes192-wrap-pad        -id-aes256-wrap            -id-aes256-wrap-
pad
-id-smime-alg-CMS3DESwrap  -rc2                       -rc2-128
-rc2-40                    -rc2-40-cbc                -rc2-64
-rc2-64-cbc                -rc2-cbc                   -rc2-cfb
-rc2-ecb                   -rc2-ofb                   -rc4
-rc4-40                    -seed                      -seed-cbc
-seed-cfb                  -seed-ecb                  -seed-ofb
-sm4                       -sm4-cbc                   -sm4-cfb
-sm4-ctr                   -sm4-ecb                   -sm4-ofb
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl sm4-cbc -K
"2851fa25211a48023794ae9515909603" -iv
"da80e405a4998c351b0717093cbe86ab" -in xlm.txt -out xlm.enc
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl#  openssl sm4-cbc -d -K
"2851fa25211a48023794ae9515909603" -iv
"da80e405a4998c351b0717093cbe86ab" -in xlm.enc -out xlm2.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl#  diff xlm.txt xlm2.txt
```

- 非对称算法
  - RSA

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl genpkey -
algorithm RSA -out private_key.pem
..+...........+.+.................+..........+......+..+.......+..
+.+..+..........+..+.........+.+........+.......+........+........
....+.+.........+......+++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++*......+...+....+.....+.++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++*..+...+..+..........
...+......+..+...+....+..+.+..+.......+......+..+.+..+...+....+...
..+......+....+......+...........+......+.........+...+.......+...
+......+......+.....+.+.................+.+......+........+...+...
+.+......+............+..+.+.....+...............+...+.+......+..+
.......+...+..+....+..............+..........+..+...+.......+.....
+.+...............++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++
.........+......+........+......++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++*.......+.....+.+.....+...+.......+
.........+++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++*..+....+...........+......+...+.......+...+..............
.......+.....+....+..+..........+...+...........+.........+......+
.+...+..+....+...+........+...+.+..................+..+...........
..+..+....+..+...+.........+..........+..+.+..+.............+.....
...+......+.+.........+...+...+...+..............+.+..+.........+.
.....+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# ls
123.bin  123.txt  bitstring.der  bitstring.pem  private_key.pem
r1.bin  r2.bin  xlm.b64  xlm.enc  xlm.txt  xlm2.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# cat
private_key.pem
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDEzWQVBetArGXG
lsafji5/vk5ezOo6UhQ4tYVI7FNwp0bbv2i4Xfe5bBeghUyCDSvr6O7olkU+g6pU
Nbgk+3zkkbVKyGdSTXgIn25SrpIbZho41mMB07CbWOpOECdxq0htyIwD/DZLWLuE
X/kjTtKVlqEis/yjhLXwYNk5Vuo8nNF4ncMiKZ3CXG/aoYL45pE9t+YXdmGWOLXi
dBlAd93iQ1TsZ+WJm7G8GG+BMvGnhdXw3eEEOl3KIzoT3VTiQquMB+FO1PTB/OJK
C4necAwYEfYHYJwaSqxZt5D6Ak8eLFK5JYJmgD1MOK4KnUZvFbvmNkBbOldu1z4V
DN/SBpp7AgMBAAECggEABMoGF6LgvN6qeAuJmC69aptmzH91Fb097mawcHwGcx/c
UBMCM+EJhb11J7wGsXArnkk5WHdLsRVC1bDG+JNsJc3L7nQeW440umpSz3fzE0VH
+b5x2K9eRVwuZimWJCR/FeOC+20tpGYWOMbzHk0ClTmoqy+xGV5B5TBzfiXje7Jp
qJDFd1E8ZkKdH4J9twA7I5WRbO88H3rPSJDhQrU47FPERvNNICbM59vLoHNpxVBX
1pBgjNTcDwN9wTtys3OBawUTcK1KrgoesV3+k6hDa6IN8mEy8BEbQ70DIYgc0V31
Sm6HN9Zkh/6Mxdvr4nB8FOAFDBnKI5cSyFpEsjaAiQKBgQDbpLsfoTrZuFswEkZ3
uJAPm75CtZmVxEeXHqp02+UBafzYwy++WfzKMpp2CXW4bgd9njSrwfULe6VVbhj9
c1Kn3Htn65KjKIEQflyBdwBhLLlb3BxvsUXlI6ij4mPc63ygCJA5WdDzfHBcgDTT
yF9hPKRVYovrjBKsq22in6JV8wKBgQDlYMc2NAMQNKWqWNSglpJbdYFNKCFhvIzc
aRGOM1L0MRvFNyv+t1nRoKeaxqCkd8h65xsar7SNNd72slyhcs9zPe3xJbdwm07q
1q0JDGvUWYTrUlRBLnL5bMAozEv2vGS8xD+7VVrgcxH0G1oD/wS14tUAD2u9t0lq
2LgjaO+jWQKBgQDPI6XNiJIlrfVhenq2gXprHefqpbT4Ryl03VjH6HEqSjhIfJtU
Gy2JyvtcgkNg8XNjBoaJzNs6PxuHW9N5gv7ai9ZeBQ4/jP1a/rBi8EWNX05X0VeI
BljyZhSuqdygBf18N1c8nvWuCxc0RTyM1hUNcNFSLSPjujAKY4l7qqy//QKBgEg6
omB6HmDTAzvR/xqWb33nUZEXSvO46O5bE5EgrkWA1UrT5cGuwNTW7xA47cr8gR/a
eFl97K/uv8gVQEACpDqYzL177/jAnygp85D+3VGf4tArO6bO1pueWCBAvMb0ahBb
B+qYpSY6dfPVTRInErentwTu1jGGbtL7bXiRCaz5AoGBALMfE3fIloCdt+bzbSkB
a76UgfQNi1D/oE4XXY6oTMpkqDSegdV1S7CCfMauAbgdfMmmcQ5PNRg3UqoD+Bdg
E5X6ZElpAd6k/Po+L21FEy1BNU6X3q3Z8vDuGj0Rl/JB6ziIie4rSR70YUB7k44W
uREJTVh50FOvmGNoRmvpQjM7
-----END PRIVATE KEY-----
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl asn1parse
-inform PEM -in private_key.pem
    0:d=0  hl=4 l=1214 cons: SEQUENCE
    4:d=1  hl=2 l=   1 prim: INTEGER           :00
    7:d=1  hl=2 l=  13 cons: SEQUENCE
    9:d=2  hl=2 l=   9 prim: OBJECT            :rsaEncryption
20:d=2  hl=2 l=   0 prim: NULL
22:d=1  hl=4 l=1192 prim: OCTET STRING      [HEX
DUMP]:308204A40201000282010100C4CD641505EB40AC65C696C69F8E2E7FBE4E
5ECCEA3A521438B58548EC5370A746DBBF68B85DF7B96C17A0854C820D2BEBE8EE
E896453E83AA5435B824FB7CE491B54AC867524D78089F6E52AE921B661A38D663
```

```
01D3B09B58EA4E102771AB486DC88C03FC364B58BB845FF9234ED29596A122B3FC
A384B5F060D93956EA3C9CD1789DC322299DC25C6FDAA182F8E6913DB7E6177661
9638B5E274194077DDE24354EC67E5899BB1BC186F8132F1A785D5F0DDE1043A5D
CA233A13DD54E242AB8C07E14ED4F4C1FCE24A0B89DE700C1811F607609C1A4AAC
59B790FA024F1E2C52B9258266803D4C38AE0A9D466F15BBE636405B3A576ED73E
150CDFD2069A7B020301000102820100004CA0617A2E0BCDEAA780B89982EBD6A9B
66CC7F7515BD3DEE66B0707C06731FDC50130233E10985BD7527BC06B1702B9E49
3958774BB11542D5B0C6F8936C25CDCBEE741E5B8E34BA6A52CF77F3134547F9BE
71D8AF5E455C2E66299624247F15E382FB6D2DA4661638C6F31E4D029539A8AB2F
B1195E41E530737E25E37BB269A890C577513C66429D1F827DB7003B2395916CEF
3C1F7ACF4890E142B538EC53C446F34D2026CCE7DBCBA07369C55057D690608CD4
DC0F037DC13B72B373816B051370AD4AAE0A1EB15DFE93A8436BA20DF26132F011
1B43BD0321881CD15DF54A6E8737D66487FE8CC5DBEBE2707C14E0050C19CA2397
12C85A44B236808902818100DBA4BB1FA13AD9B85B30124677B8900F9BBE42B599
95C447971EAA74DBE50169FCD8C32FBE59FCCA329A760975B86E077D9E34ABC1F5
0B7BA5556E18FD7352A7DC7B67EB92A32881107E5C817700612CB95BDC1C6FB145
E523A8A3E263DCEB7CA008903959D0F37C705C8034D3C85F613CA455628BEB8C12
ACAB6DA29FA255F302818100E560C73634031034A5AA58D4A096925B75814D2821
61BC8CDC69118E3352F4311BC5372BFEB759D1A0A79AC6A0A477C87AE71B1AAFB4
8D35DEF6B25CA172CF733DEDF125B7709B4EEAD6AD090C6BD45984EB5254412E72
F96CC028CC4BF6BC64BCC43FBB555AE07311F41B5A03FF04B5E2D5000F6BBDB749
6AD8B82368EFA35902818100CF23A5CD889225ADF5617A7AB6817A6B1DE7EAA5B4
F8472974DD58C7E8712A4A38487C9B541B2D89CAFB5C824360F17363068689CCDB
3A3F1B875BD37982FEDA8BD65E050E3F8CFD5AFEB062F0458D5F4E57D157880658
F26614AEA9DCA005FD7C37573C9EF5AE0B1734453C8CD6150D70D1522D23E3BA30
0A63897BAAACBFFD028180483AA2607A1E60D3033BD1FF1A966F7DE75191174AF3
B8E8EE5B139120AE4580D54AD3E5C1AEC0D4D6EF1038EDCAFC811FDA78597DECAF
EEBFC815404002A43A98CCBD7BEFF8C09F2829F390FEDD519FE2D02B3BA6CED69B
9E582040BCC6F46A105B07EA98A5263A75F3D54D122712B7A7B704EED631866ED2
FB6D789109ACF902818100B31F1377C896809DB7E6F36D29016BBE9481F40D8B50
FFA04E175D8EA84CCA64A8349E81D5754BB0827CC6AE01B81D7CC9A6710E4F3518
3752AA03F817601395FA64496901DEA4FCFA3E2F6D45132D41354E97DEADD9F2F0
EE1A3D1197F241EB388889EE2B491EF461407B938E16B911094D5879D053AF9863
68466BE942333B
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl genpkey -
help
Usage: genpkey [options]

General options:
-help                Display this summary
-engine val          Use engine, possibly a hardware device
-paramfile infile    Parameters file
-algorithm val       The public key algorithm
-quiet               Do not output status while generating keys
-pkeyopt val         Set the public key algorithm option as
opt:value
-config infile       Load a configuration file (this may load
modules)

Output options:
-out outfile         Output file
-outform PEM|DER     output format (DER or PEM)
-pass val            Output file pass phrase source
-genparam            Generate parameters, not key
```

```
-text               Print the in text
-*                  Cipher to use to encrypt the key

Provider options:
-provider-path val  Provider load path (must be before 'provider'
argument if required)
-provider val       Provider to load (can be specified multiple
times)
-propquery val      Property query used when fetching algorithms
Order of options may be important!  See the documentation.
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git commit -m
"create RSA keys"
[master bf6fe6b] create RSA keys
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are
accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to
edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit
with:

    git commit --amend --reset-author

2 files changed, 30 insertions(+)
create mode 100644 shiyan1-1/openssl/private_key.pem
create mode 100644 shiyan1-1/openssl/xlm.enc
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git commit --amend
[master 11df9cc] finish enc command and create RSA keys
Date: Sun Oct 13 11:13:52 2024 +0800
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are
accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to
edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit
with:

    git commit --amend --reset-author

2 files changed, 30 insertions(+)
create mode 100644 shiyan1-1/openssl/private_key.pem
create mode 100644 shiyan1-1/openssl/xlm.enc
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl rsa -
pubout -in privatekey.pem -out publickey.pem
Could not open file or uri for loading private key from
privatekey.pem
80DBA3D3017F0000:error:16000069:STORE
routines:ossl_store_get0_loader_int:unregistered
scheme:../crypto/store/store_register.c:237:scheme=file
80DBA3D3017F0000:error:80000002:system library:file_open:No such
file or
directory:../providers/implementations/storemgmt/file_store.c:267:
calling stat(privatekey.pem)
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# ls
123.bin  123.txt  bitstring.der  bitstring.pem  private_key.pem
r1.bin  r2.bin  xlm.b64  xlm.enc  xlm.txt  xlm2.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl#  openssl rsa -
pubout -in private_key.pem -out publickey.pem
writing RSA key
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# mv private_key.pem
privatekey.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# cat publickey.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxM1kFQXrQKxlxpbGn44u
f75OXszqOlIUOLWFSOxTcKdG279ouF33uWwXoIVMgg0r6+ju6JZFPoOqVDW4JPt8
5JG1SshnUk14CJ9uUq6SG2YaONZjAdOwm1jqThAncatIbciMA/w2S1i7hF/5I07S
lZahIrP8o4S18GDZOVbqPJzReJ3DIimdwlxv2qGC+OaRPbfmF3Zhlji14nQZQHfd
4kNU7GfliZuxvBhvgTLxp4XV8N3hBDpdyiM6E91U4kKrjAfhTtT0wfziSguJ3nAM
GBH2B2CcGkqsWbeQ+gJPHixSuSWCZoA9TDiuCp1GbxW75jZAWzpXbtc+FQzf0gaa
ewIDAQAB
-----END PUBLIC KEY-----
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl pkeyutl -
encrypt -inkey publickey.pem -pubin -in xlm.txt -out xlmrsaenc.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl pkeyutl -
decrypt -inkey privatekey.pem -in xlmrsaenc.bin -out xlmrsadec.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# ls
123.bin  bitstring.der  privatekey.pem  r1.bin  xlm.b64  xlm.txt
xlmrsadec.txt
123.txt  bitstring.pem  publickey.pem   r2.bin  xlm.enc  xlm2.txt
xlmrsaenc.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# diff xlm.txt
xlmrsadec.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl pkeyutl -
help
Usage: pkeyutl [options]

General options:
-help                      Display this summary
-engine val                Use engine, possibly a hardware device
-engine_impl               Also use engine given by -engine for
crypto operations
-sign                      Sign input data with private key
-verify                    Verify with public key
-encrypt                   Encrypt input data with public key
-decrypt                   Decrypt input data with private key
```

```
 -derive                   Derive shared secret
 -config infile            Load a configuration file (this may load
modules)

Input options:
 -in infile                Input file - default stdin
 -rawin                    Indicate the input data is in raw form
 -pubin                    Input is a public key
 -inkey val                Input private key file
 -passin val               Input file pass phrase source
 -peerkey val              Peer key file used in key derivation
 -peerform PEM|DER|ENGINE  Peer key format (DER/PEM/P12/ENGINE)
 -certin                   Input is a cert with a public key
 -rev                      Reverse the order of the input buffer
 -sigfile infile           Signature file (verify operation only)
 -keyform PEM|DER|ENGINE   Private key format (ENGINE, other values
ignored)

Output options:
 -out outfile              Output file - default stdout
 -asn1parse                asn1parse the output data
 -hexdump                  Hex dump output
 -verifyrecover            Verify with public key, recover original
data

Signing/Derivation options:
 -digest val               Specify the digest algorithm when
signing the raw input data
 -pkeyopt val              Public key options as opt:value
 -pkeyopt_passin val       Public key option that is read as a
passphrase argument opt:passphrase
 -kdf val                  Use KDF algorithm
 -kdflen +int              KDF algorithm output length

Random state options:
 -rand val                 Load the given file(s) into the random
number generator
 -writerand outfile        Write random data to the specified file

Provider options:
 -provider-path val        Provider load path (must be before
'provider' argument if required)
 -provider val             Provider to load (can be specified
multiple times)
 -propquery val            Property query used when fetching
algorithms
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl dgst -
sha256 -sign privatekey.pem -out xlm.sig xlm.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl dgst -
sha256 -verify publickey.pem -signature xlm.sig xlm.txt
Verified OK
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl pkeyutl -
sign -inkey privatekey.pem -in xlm.txt -out xlmrsa.sig
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl pkeyutl -
```

```
verify -in xlm.txt -sigfile xlmrsa.sig -inkey privatekey.pem
Signature Verified Successfully
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git commit -m
"finish RSA command"
[master 3bd6f65] finish RSA command
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are
accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to
edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit
with:

    git commit --amend --reset-author

6 files changed, 11 insertions(+)
rename shiyan1-1/openssl/{private_key.pem => privatekey.pem}
(100%)
create mode 100644 shiyan1-1/openssl/publickey.pem
create mode 100644 shiyan1-1/openssl/xlm.sig
create mode 100644 shiyan1-1/openssl/xlmrsa.sig
create mode 100644 shiyan1-1/openssl/xlmrsadec.txt
create mode 100644 shiyan1-1/openssl/xlmrsaenc.bin
```

- SM2

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl ecparam -
genkey -name SM2 -out sm2private_key.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# cat
sm2private_key.pem
-----BEGIN SM2 PARAMETERS-----
BggqgRzPVQGCLQ==
-----END SM2 PARAMETERS-----
-----BEGIN PRIVATE KEY-----
MIGIAgEAMBQGCCqBHM9VAYItBggqgRzPVQGCLQRtMGsCAQEEIApgfY1Px4JplNNE
w0C4gdc2axdRbLMseWa+o5D1j1/ZoUQDQgAEUorFPGit0LSUcLdMoWhAAL2m+FnS
J94hsmu3bQwOSONARKhMhXNsIaLiOpvwM52Z2XlC6Gas9+d0f5XrE4uabw==
-----END PRIVATE KEY-----
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl asn1parse
-inform PEM -in sm2private_key.pem
    0:d=0  hl=2 l=   8 prim: OBJECT            :sm2
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl base64 -d
-in sm2privatekey.pem -out sm2privatekey.der
Can't open "sm2privatekey.pem" for reading, No such file or
directory
```

```
801BBCEB9B7F0000:error:80000002:system library:BIO_new_file:No
such file or directory:../crypto/bio/bss_file.c:67:calling
fopen(sm2privatekey.pem, r)
801BBCEB9B7F0000:error:10000080:BIO routines:BIO_new_file:no such
file:../crypto/bio/bss_file.c:75:
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl#  openssl base64 -d
-in sm2private_key.pem -out sm2private_key.der
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl asn1parse
-inform DER -in sm2private_key.der
    0:d=0  hl=2 l=   8 prim: OBJECT            :sm2
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl pkey -in
sm2private_key.pem -text -noout
Private-Key: (256 bit)
priv:
    0a:60:7d:8d:4f:c7:82:69:94:d3:44:c3:40:b8:81:
    d7:36:6b:17:51:6c:b3:2c:79:66:be:a3:90:f5:8f:
    5f:d9
pub:
    04:52:8a:c5:3c:68:ad:d0:b4:94:70:b7:4c:a1:68:
    40:00:bd:a6:f8:59:d2:27:de:21:b2:6b:b7:6d:0c:
    0e:48:e3:40:44:a8:4c:85:73:6c:21:a2:e2:3a:9b:
    f0:33:9d:99:d9:79:42:e8:66:ac:f7:e7:74:7f:95:
    eb:13:8b:9a:6f
ASN1 OID: SM2
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl ecparam -
help
Usage: ecparam [options]

General options:
-help                Display this summary
-list_curves         Prints a list of all curve 'short names'
-engine val          Use engine, possibly a hardware device
-genkey              Generate ec key
-in infile           Input file  - default stdin
-inform PEM|DER      Input format - default PEM (DER or PEM)
-out outfile         Output file - default stdout
-outform PEM|DER     Output format - default PEM

Output options:
-text                Print the ec parameters in text form
-noout               Do not print the ec parameter
-param_enc val       Specifies the way the ec parameters are
encoded

Parameter options:
-check               Validate the ec parameters
-check_named         Check that named EC curve parameters have not
been modified
-no_seed             If 'explicit' parameters are chosen do not use
the seed
-name val            Use the ec parameters with specified 'short
name'
-conv_form val       Specifies the point conversion form
```

```
Random state options:
-rand val           Load the given file(s) into the random number generator
-writerand outfile  Write random data to the specified file

Provider options:
-provider-path val  Provider load path (must be before 'provider' argument if required)
-provider val       Provider to load (can be specified multiple times)
-propquery val      Property query used when fetching algorithms
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl ec -in sm2private_key.pem -pubout -out sm2public_key.pem
read EC key
writing EC key
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# cat sm2public_key.pem
-----BEGIN PUBLIC KEY-----
MFowFAYIKoEcz1UBgi0GCCqBHM9VAYItA0IABFKKxTxordC0lHC3TKFoQAC9pvhZ
0ifeIbJrt20MDkjjQESoTIVzbCGi4jqb8DOdmdl5QuhmrPfndH+V6xOLmm8=
-----END PUBLIC KEY-----
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl asn1parse -inform PEM -in sm2public_key.pem
    0:d=0  hl=2 l=  90 cons: SEQUENCE
    2:d=1  hl=2 l=  20 cons: SEQUENCE
    4:d=2  hl=2 l=   8 prim: OBJECT            :sm2
14:d=2  hl=2 l=   8 prim: OBJECT            :sm2
24:d=1  hl=2 l=  66 prim: BIT STRING
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl ec -help
Usage: ec [options]

General options:
-help               Display this summary
-engine val         Use engine, possibly a hardware device

Input options:
-in val             Input file
-inform format      Input format (DER/PEM/P12/ENGINE)
-pubin              Expect a public key in input file
-passin val         Input file pass phrase source
-check              check key consistency
-*                  Any supported cipher
-param_enc val      Specifies the way the ec parameters are encoded
-conv_form val      Specifies the point conversion form

Output options:
-out outfile        Output file
-outform PEM|DER    Output format - DER or PEM
-noout              Don't print key out
-text               Print the key
-param_out          Print the elliptic curve parameters
-pubout             Output public key, not private
-no_public          exclude public key from private key
```

```
-passout val        Output file pass phrase source

Provider options:
-provider-path val  Provider load path (must be before 'provider'
argument if required)
-provider val       Provider to load (can be specified multiple
times)
-propquery val      Property query used when fetching algorithms
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl pkeyutl -
encrypt -pubin -inkey sm2public_key.pem -in xlm.txt -out
xlmsm2enc.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl pkeyutl -
decrypt -inkey sm2private_key.pem -in xlmsm2enc.bin -out
xlmsm2dec.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# diff xlm.txt
xlmsm2dec.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl sm3 -sign
sm2private_key.pem -out xlmsm2.sig xlm.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl sm3 -
verify sm2public_key.pem -signature xlmsm2.sig xlm.txt
Verified OK
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl pkeyutl -
sign -in xlm.txt -inkey sm2private_key.pem -out xlmsm2.sig -rawin
-digest sm3
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# od -tx1 xlmsm2.sig
0000000 30 45 02 20 37 19 03 7e f7 8d 55 f9 74 a8 f3 75
0000020 29 de bc 3e c5 f8 64 5e ea d1 d6 f6 3e 5d 69 41
0000040 a7 97 12 dc 02 21 00 9f 4d ea 77 0c 45 a8 f6 a1
0000060 53 01 ab b3 ba 0a d7 15 e7 e4 55 f4 8a 91 a4 b4
0000100 97 b3 5e 09 d8 5a a4
0000107
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl pkeyutl -
verify -in xlm.txt -inkey sm2private_key.pem -sigfile xlmsm2.sig -
rawin -digest sm3
Signature Verified Successfully
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git commit -m
"finish sm2 command"
[master 264f8f9] finish sm2 command
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are
accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to
edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit
with:

    git commit --amend --reset-author
```

```
 6 files changed, 14 insertions(+)
 create mode 100644 shiyan1-1/openssl/sm2private_key.der
 create mode 100644 shiyan1-1/openssl/sm2private_key.pem
 create mode 100644 shiyan1-1/openssl/sm2public_key.pem
 create mode 100644 shiyan1-1/openssl/xlmsm2.sig
 create mode 100644 shiyan1-1/openssl/xlmsm2dec.txt
 create mode 100644 shiyan1-1/openssl/xlmsm2enc.bin
```

- 其他命令

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# openssl list -commands
asn1parse         ca                ciphers           cmp
cms               crl               crl2pkcs7         dgst
dhparam           dsa               dsaparam          ec
ecparam           enc               engine            errstr
fipsinstall       gendsa            genpkey           genrsa
help              info              kdf               list
mac               nseq              ocsp              passwd
pkcs12            pkcs7             pkcs8             pkey
pkeyparam         pkeyutl           prime             rand
rehash            req               rsa               rsautl
s_client          s_server          s_time            sess_id
smime             speed             spkac             srp
storeutl          ts                verify            version
x509
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git commit -m "finish openssl commands"
On branch master
nothing to commit, working tree clean
```

- git-log

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# git log
commit 264f8f95efec51e6dfeb033faa28bb524d0e29ef (HEAD -> master)
Author: root <root@Youer>
Date:   Sun Oct 13 11:34:27 2024 +0800

    finish sm2 command

commit 3bd6f65fb20217bd47645c78a54886cbf3845a36
Author: root <root@Youer>
Date:   Sun Oct 13 11:24:50 2024 +0800

    finish RSA command

commit 11df9ccb169626e278c5ff382060888c4077c8fb
```

```
Author: root <root@Youer>
Date:   Sun Oct 13 11:13:52 2024 +0800

    finish enc command and create RSA keys

commit 0f82610bece47c5171b53bcf5a4e5165931e2962
Author: root <root@Youer>
Date:   Sun Oct 13 10:50:16 2024 +0800

    finish asn1parse command

commit 9c2859a5bcc7796fcbe0142f4de89d23fba87e25
Author: root <root@Youer>
Date:   Sun Oct 13 10:47:08 2024 +0800

    finish base64 command

commit 3dce8b6d62515695739d6d0d226531c292ae06c5
Author: root <root@Youer>
Date:   Sun Oct 13 10:27:09 2024 +0800

    Input and Output of Data in Different Bases

```

---

2. 参考云班课相关教学视频，在 Ubuntu或openEuler中（推荐 openEuler）中实践课程课程思维导图中GmSSL相关内容，使用Markdown记录详细记录实践过程，每完成一项git commit 一次。（5‘）

- 实践过程
  - 初始准备

    ```
    root@Youer:~/shiyan/shiyan01/shiyan1-1/openssl# cd ..
    root@Youer:~/shiyan/shiyan01/shiyan1-1# ls
    openssl
    root@Youer:~/shiyan/shiyan01/shiyan1-1# mkdir gmssl
    root@Youer:~/shiyan/shiyan01/shiyan1-1# cd gmssl
    root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# git add .
    root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# git commit -m "start
    gmssl commands"
    On branch master
    nothing to commit, working tree clean
    ```

  - 基础
    - help and version

      ```
      root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl help
      usage: gmssl command [options]
      command -help
      ```

```
Commands:
help              Print this help message
version           Print version
rand              Generate random bytes
sm2keygen         Generate SM2 keypair
sm2sign           Generate SM2 signature
sm2verify         Verify SM2 signature
sm2encrypt        Encrypt with SM2 public key
sm2decrypt        Decrypt with SM2 private key
sm3               Generate SM3 hash
sm3hmac           Generate SM3 HMAC tag
sm3_pbkdf2        Hash password into key using PBKDF2 algoritm
sm3xmss_keygen    Generate SM3-XMSS keypair
sm4_ecb           Encrypt or decrypt with SM4 ECB
sm4_cbc           Encrypt or decrypt with SM4 CBC
sm4_ctr           Encrypt or decrypt with SM4 CTR
sm4_cfb           Encrypt or decrypt with SM4 CFB
sm4_ofb           Encrypt or decrypt with SM4 OFB
sm4_ccm           Encrypt or decrypt with SM4 CCM
sm4_gcm           Encrypt or decrypt with SM4 GCM
sm4_xts           Encrypt or decrypt with SM4 XTS
sm4_cbc_sm3_hmac  Encrypt or decrypt with SM4 CBC with SM3-HMAC
sm4_ctr_sm3_hmac  Encrypt or decrypt with SM4 CTR with SM3-HMAC
sm4_cbc_mac       Generate SM4 CBC-MAC
ghash             Generate GHASH
zuc               Encrypt or decrypt with ZUC
sm9setup          Generate SM9 master secret
sm9keygen         Generate SM9 private key
sm9sign           Generate SM9 signature
sm9verify         Verify SM9 signature
sm9encrypt        SM9 public key encryption
sm9decrypt        SM9 decryption
reqgen            Generate certificate signing request (CSR)
reqsign           Generate certificate from CSR
reqparse          Parse and print a CSR
crlget            Download the CRL of given certificate
crlgen            Sign a CRL with CA certificate and private key
crlverify         Verify a CRL with issuer's certificate
crlparse          Parse and print CRL
certgen           Generate a self-signed certificate
certparse         Parse and print certificates
certverify        Verify certificate chain
certrevoke        Revoke certificate and output RevokedCertificate
record
cmsparse          Parse CMS (cryptographic message syntax) file
cmsencrypt        Generate CMS EnvelopedData
cmsdecrypt        Decrypt CMS EnvelopedData
cmssign           Generate CMS SignedData
cmsverify         Verify CMS SignedData
sdfinfo           Print SDF device info
sdfdigest         Generate SM3 hash with SDF device
sdfexport         Export SM2 signing public key from SDF device
sdfsign           Generate SM2 signature with SDF internal private
```

```
key
sdfencrypt        SM2/SM4-CBC hybrid encryption with SDF device
sdfdecrypt        SM2/SM4-CBC hybrid decryption with SDF device
sdftest           Test vendor's SDF library and device
tlcp_client       TLCP client
tlcp_server       TLCP server
tls12_client      TLS 1.2 client
tls12_server      TLS 1.2 server
tls13_client      TLS 1.3 client
tls13_server      TLS 1.3 server

run `gmssl <command> -help` to print help of the given command


root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl version
GmSSL 3.1.2 Dev
```

- sm3

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl sm3 -help
usage: sm3 [-hex|-bin] [-pubkey pem [-id str]] [-in file|-in_str
str] [-out file]
Options

    -hex                   Output hash value as hex string (by
default)
    -bin                   Output hash value as binary
    -pubkey pem            Signer's SM2 public key
                        When `-pubkey` is specified, hash with SM2
Z value
    -id str                SM2 Signer's ID string
    -id_hex hex            SM2 Signer's ID in hex format
                        `-id` and `-id_hex` should be used with `-
pubkey`
                        `-id` and `-id_hex` should not be used
together
                        If `-pubkey` is specified without `-id` or
`id_hex`,
                        the default ID string '1234567812345678'
is used
    -in_str str            To be hashed string
    -in file | stdin       To be hashed file path
                        `-in_str` and `-in` should not be used
together
                        If neither `-in` nor `-in_str` specified,
read from stdin
    -out file | stdout     Output file path. If not specified,
output to stdout

Examples

    gmssl sm3 -in_str abc
    gmssl sm3 -in_str abc -bin

    gmssl sm3 -in /path/to/file

    gmssl sm3 -pubkey sm2pubkey.pem -id alice -in /path/to/file -bin

When reading from stdin, make sure the trailing newline character is removed

Linux/Mac:
    echo -n abc | gmssl sm3

Windows:
    C:\> echo |set/p="abc" | gmssl sm3

root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo -n "xlm" | gmssl sm3
99f620e94508ee9445bf0722bac8d9d9942cd1a9821f99b2e9e416960e926596
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo -n "xlm" | gmssl sm3 -hex
99f620e94508ee9445bf0722bac8d9d9942cd1a9821f99b2e9e416960e926596
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo -n "xlm" | gmssl sm3 -bin
��
��E�"��̡�,ʍ��������e�root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo -n "xlm" | gmssl sm3 -bin | od -tx1
0000000 99 f6 20 e9 45 08 ee 94 45 bf 07 22 ba c8 d9 d9
0000020 94 2c d1 a9 82 1f 99 b2 e9 e4 16 96 0e 92 65 96
0000040
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo -n "xlm" > xlm.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl#  od -tx1 -tc xlm.txt
0000000  78  6c  6d
          x   l   m
0000003
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl sm3 -in xlm.txt -out xlm.sm3
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl#  cat xlm.sm3
99f620e94508ee9445bf0722bac8d9d9942cd1a9821f99b2e9e416960e926596
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl sm2keygen -pass 1234 -out sm2.pem -pubout sm2pub.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl#  ls
sm2.pem  sm2pub.pem  xlm.sm3  xlm.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo -n "xlm" | gmssl sm3 -pubkey sm2pub.pem -id 1234567812345678
5f555522761c81d92e98d301eb55f93b53272c463d632b3867c2a6f6ec7d37a8
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl sm3hmac -help
usage: sm3hmac -key hex [-in file | -in_str str] [-bin|-hex] [-out file]
Options

    -key hex               Hex string of the MAC key
    -in_str str            Input as text string
    -in file | stdin       Input file path
                        `-in_str` and `-in` should not be used
together
                        If neither `-in` nor `-in_str` specified,
read from stdin
    -hex                   Output MAC-tag as hex string (by
default)
    -bin                   Output MAC-tag as binary
                        `-hex` and `-bin` should not be used
together
    -out file | stdout     Output file path. If not specified,
output to stdout

Examples

    KEY_HEX=`gmssl rand -outlen 16 -hex`
    gmssl sm3hmac -key $KEY_HEX -in_str abc

    gmssl sm3hmac -key $KEY_HEX -in_str abc -bin

    gmssl sm3hmac -key $KEY_HEX -in /path/to/file

When reading from stdin, make sure the trailing newline character
is removed

Linux/Mac:
    echo -n abc | gmssl sm3hmac -key $KEY_HEX

Windows:
    C:\> echo |set/p="abc" | gmssl sm3hmac -key
11223344556677881122334455667788


root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl rand -help
usage: rand [-hex] [-rdrand|-rdseed] -outlen num [-out file]
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl rand -hex -
outlen 16
5CA709DE420CD9603C903E2B16B90834
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo -n "xlm" |
gmssl sm3hmac -key 5CA709DE420CD9603C903E2B16B90834
54b9bb8ee1b03f9e0005233d9d5745a321c04f13288071e9f47f1306414449a0
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# git commit -m
"finish gmssl sm3 command"
[master 8efe684] finish gmssl sm3 command
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are
accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to
edit
your configuration file:
    git config --global --edit

After doing this, you may fix the identity used for this commit
with:

    git commit --amend --reset-author

4 files changed, 14 insertions(+)
create mode 100644 shiyan1-1/gmssl/sm2.pem
create mode 100644 shiyan1-1/gmssl/sm2pub.pem
create mode 100644 shiyan1-1/gmssl/xlm.sm3
create mode 100644 shiyan1-1/gmssl/xlm.txt
```

- sm4

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl sm4 -help
gmssl: illegal option 'sm4'
usage: gmssl command [options]
command -help

Commands:
help                 Print this help message
version              Print version
rand                 Generate random bytes
sm2keygen            Generate SM2 keypair
sm2sign              Generate SM2 signature
sm2verify            Verify SM2 signature
sm2encrypt           Encrypt with SM2 public key
sm2decrypt           Decrypt with SM2 private key
sm3                  Generate SM3 hash
sm3hmac              Generate SM3 HMAC tag
sm3_pbkdf2           Hash password into key using PBKDF2 algoritm
sm3xmss_keygen       Generate SM3-XMSS keypair
sm4_ecb              Encrypt or decrypt with SM4 ECB
sm4_cbc              Encrypt or decrypt with SM4 CBC
sm4_ctr              Encrypt or decrypt with SM4 CTR
sm4_cfb              Encrypt or decrypt with SM4 CFB
sm4_ofb              Encrypt or decrypt with SM4 OFB
sm4_ccm              Encrypt or decrypt with SM4 CCM
sm4_gcm              Encrypt or decrypt with SM4 GCM
sm4_xts              Encrypt or decrypt with SM4 XTS
sm4_cbc_sm3_hmac     Encrypt or decrypt with SM4 CBC with SM3-HMAC
sm4_ctr_sm3_hmac     Encrypt or decrypt with SM4 CTR with SM3-HMAC
sm4_cbc_mac          Generate SM4 CBC-MAC
ghash                Generate GHASH
zuc                  Encrypt or decrypt with ZUC
sm9setup             Generate SM9 master secret
sm9keygen            Generate SM9 private key
sm9sign              Generate SM9 signature
sm9verify            Verify SM9 signature
sm9encrypt           SM9 public key encryption
```

```
sm9decrypt        SM9 decryption
reqgen            Generate certificate signing request (CSR)
reqsign           Generate certificate from CSR
reqparse          Parse and print a CSR
crlget            Download the CRL of given certificate
crlgen            Sign a CRL with CA certificate and private key
crlverify         Verify a CRL with issuer's certificate
crlparse          Parse and print CRL
certgen           Generate a self-signed certificate
certparse         Parse and print certificates
certverify        Verify certificate chain
certrevoke        Revoke certificate and output RevokedCertificate
record
cmsparse          Parse CMS (cryptographic message syntax) file
cmsencrypt        Generate CMS EnvelopedData
cmsdecrypt        Decrypt CMS EnvelopedData
cmssign           Generate CMS SignedData
cmsverify         Verify CMS SignedData
sdfinfo           Print SDF device info
sdfdigest         Generate SM3 hash with SDF device
sdfexport         Export SM2 signing public key from SDF device
sdfsign           Generate SM2 signature with SDF internal private
key
sdfencrypt        SM2/SM4-CBC hybrid encryption with SDF device
sdfdecrypt        SM2/SM4-CBC hybrid decryption with SDF device
sdftest           Test vendor's SDF library and device
tlcp_client       TLCP client
tlcp_server       TLCP server
tls12_client      TLS 1.2 client
tls12_server      TLS 1.2 server
tls13_client      TLS 1.3 client
tls13_server      TLS 1.3 server

run `gmssl <command> -help` to print help of the given command


root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl rand -outlen
16 -out key.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl rand -outlen
16 -out iv.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# ls
iv.bin  key.bin  sm2.pem  sm2pub.pem  xlm.sm3  xlm.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl#  od -tx1 key.bin
0000000 44 bf 1d 1b f9 11 28 18 d9 2b 6d 45 c7 46 98 1e
0000020
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# od -tx1 iv.bin
0000000 fc db 1d ab 17 a2 75 46 cd ca e8 19 b6 fb c3 80
0000020
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo -n "xlm" |
gmssl sm4_cbc -encrypt -key $(xxd -p -c 32 key.bin) -iv $(xxd -p -
c 32 iv.bin) -out xlmsm4.cbcgmssl sm4_cbc -help
gmssl sm4_cbc: illegal option `sm4_cbc`
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo -n "xlm" |
gmssl sm4_cbc -encrypt -key $(xxd -p -c 32 key.bin) -iv $(
xxd -p -c 32 iv.bin) -out xlmsm4.cbcgmssl
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# ls
iv.bin  key.bin  sm2.pem  sm2pub.pem  xlm.sm3  xlm.txt
xlmsm4.cbcgmssl
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# mv xlmsm4.cbcgmssl
xlmsm4.cbc
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl sm4_cbc -
decrypt -key $(xxd -p -c 32 key.bin) -iv $(xxd -p -c 32 iv.
bin) -in xlmsm4.cbc
xlmroot@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# KEY=$(xxd -p -c
32 key.bin)
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo $KEY
44bf1d1bf9112818d92b6d45c746981e
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo -n "xlm" |
gmssl sm4_cbc -encrypt -key $KEY -iv $IV -out xlmsm4.cbc2
gmssl sm4_cbc: invalid IV length
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# IV=$(xxd -p -c 32
iv.bin)
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo $IV
fcdb1dab17a27546cdcae819b6fbc380
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo -n "xlm" |
gmssl sm4_cbc -encrypt -key $KEY -iv $IV -out xlmsm4.cbc2
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl sm4_cbc -
decrypt -key $KEY -iv $IV -in xlmsm4.cbc2
xlmroot@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# diff xlmsm4.cbc
xlmsm4.cbc2
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl sm4_cbc -
encrypt -key $(xxd -p -c 32 key.bin) -iv $(xxd -p -c 32 iv.
bin) -in xlm.txt -out xlmsm4.cbc3
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl sm4_cbc -
decrypt -key $(xxd -p -c 32 key.bin) -iv $(xxd -p -c 32 iv.
bin) -in xlmsm4.cbc3
xlmroot@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# diff xlmsm4.cbc
xlmsm4.cbc3
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# git commit -m
"finish gmssl sm4 command"
[master 18446d1] finish gmssl sm4 command
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are
accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to
edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit
with:

    git commit --amend --reset-author

```

```
 5 files changed, 5 insertions(+)
 create mode 100644 shiyan1-1/gmssl/iv.bin
 create mode 100644 shiyan1-1/gmssl/key.bin
 create mode 100644 shiyan1-1/gmssl/xlmsm4.cbc
 create mode 100644 shiyan1-1/gmssl/xlmsm4.cbc2
 create mode 100644 shiyan1-1/gmssl/xlmsm4.cbc3
```

- sm2

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl sm2keygen -
pass 1234 -out sm2.pem -pubout sm2pub.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# cat sm2.pem
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIBBjBhBgkqhkiG9w0BBQ0wVDA0BgkqhkiG9w0BBQwwJwQQqw+1UvbvBQ6V5stg
uw7lKgIDAQAAAgEQMAsGCSqBHM9VAYMRAjAcBggqgRzPVQFoAgQQ2DXErb0x/6kB
YAh971x13gSBoM8cZyOvIAYu7CLXJ8CVoMvYX3Yghd1JlVulxEmuT/yXDJSPB2ut
OQNr72hisw8GoAn7l2//NikCp3hyhxO/3rrwAHTOCSsNRRzqmu/O6s27TQtVMKU1
olNpECZgOLgn4x2Y6nlZAadqB/YKQdga6arng2ScOuPr3GsztVEHzNBoKojmh2y/
zIqM/G9m88Q6oCB2Ppsat52ZSjFpsPqeUVU=
-----END ENCRYPTED PRIVATE KEY-----
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# cat sm2pub.pem
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoEcz1UBgi0DQgAE6X9SpfW/nXPV+LDj1fEEf117l0F1
KZHFNv+pUCio56K3/lwtogoeUWDPavYk0DDMAf752Ry0cydiZwrONKKW6A==
-----END PUBLIC KEY-----
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# $ echo xlm | gmssl
sm2sign -key sm2.pem -pass 1234 -out sm2.sig #-id 12345
67812345678
$: command not found
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo xlm | gmssl
sm2sign -key sm2.pem -pass 1234 -out sm2.sig #-id 1234567812345678
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# od -tx1 sm2.sig
0000000 30 44 02 20 1f b2 98 53 82 be f2 3f 80 0e 45 e9
0000020 32 46 9a 6e ba c0 30 80 94 8f 13 83 c1 aa 4b 58
0000040 19 7f 70 28 02 20 59 a3 5c c0 91 b7 7a ad 85 8f
0000060 41 1a d5 d8 de b6 c9 06 83 61 9b 47 19 17 5f f0
0000100 9f 46 78 52 c8 e3
0000106
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo xlm | gmssl
sm2verify -pubkey sm2pub.pem -sig sm2.sig -id 12345678123
45678
verify : success
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# echo xlm | gmssl
sm2encrypt -pubkey sm2pub.pem -out sm2.der
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# od -tx1 sm2.der
0000000 30 6d 02 20 3f 06 ed 33 86 65 88 0d 54 fe a3 27
0000020 71 78 36 69 8e 74 27 b6 c0 da 03 51 dd 1f 43 ef
0000040 2b a7 96 43 02 21 00 d5 fb 30 57 d9 25 e1 84 f6
0000060 a0 00 93 48 1d fc 3d 9f 52 59 24 34 f3 ab bc 4e
0000100 24 6a fe 70 1e 2d ae 04 20 0b 48 d9 4f 1d ce 20
0000120 99 d5 78 e8 75 cb 1c c4 2c d1 60 b3 98 a7 3d e0
0000140 d4 d6 7c 20 e8 02 3d 3a 81 04 04 18 1c 30 08
```

```
0000157
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# gmssl sm2decrypt -
key sm2.pem -pass 1234 -in sm2.der
xlm
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# git commit -m
"finish gmssl sm2 command"
[master eab1cfa] finish gmssl sm2 command
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are
accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to
edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit
with:

    git commit --amend --reset-author

4 files changed, 11 insertions(+), 10 deletions(-)
create mode 100644 shiyan1-1/gmssl/sm2.der
rewrite shiyan1-1/gmssl/sm2.pem (81%)
create mode 100644 shiyan1-1/gmssl/sm2.sig
```

- git-log

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/gmssl# git log
commit eab1cfabc068e466edd4a6456fb6328389bc40ce (HEAD -> master)
Author: root <root@Youer>
Date:   Sun Oct 13 12:20:52 2024 +0800

    finish gmssl sm2 command

commit 18446d144cf7e079831aaeba850f26d324d403e6
Author: root <root@Youer>
Date:   Sun Oct 13 12:15:33 2024 +0800

    finish gmssl sm4 command

commit 8efe6849c073dad914f016ce316fb22bc1849d47
Author: root <root@Youer>
Date:   Sun Oct 13 12:00:22 2024 +0800

    finish gmssl sm3 command

commit 264f8f95efec51e6dfeb033faa28bb524d0e29ef
```

```
Author: root <root@Youer>
Date:   Sun Oct 13 11:34:27 2024 +0800
```

- 特殊问题：gmssl 没有直接的sm4命令，只有一些子命令。

---

3. 两人一组，在 Ubuntu或openEuler中（推荐 openEuler）中使用OpenSSL命令实现带签名的数字信封协议。使用OpenSSL时Alice发送，Bob接收。Ailice，Bob在实验中要替换为自己的8位学号+姓名。 使用Markdown记录详细记录实践过程，每完成一项git commit 一次。（10分）

- Alice,Bob生成自己的公私钥匙对，记作：（PKa，SKa），（PKb，SKb），Alice,Bob分别拥有：（PKa，SKa，PKb），（PKb，SKb，PKa），实验中把公钥文件拷贝给对方
- Alice发给Bob的明文plain.txt，内容为自己的姓名学号
- Alice：sm4 key使用gmssl rand 产生，16字节，记作k
- Alice：Sm4Enc(k,P) = C
- Alice：Sm2Enc(PKb,k) = KC
- Alice：Sm2Sign（SKa，C）= S1
- Alice： 数字信封 C||KC||S1 发给Bob
- Bob：Sm2Very（PKa，S1）
- Bob：Sm2Dec（SKb，KC）= k
- Bob：Sm4Dec（k，C）= P 我是Alice：
- Alice生成自己的公私钥匙对

```
root@Youer:~/shiyan/shiyan01/shiyan1-1# mkdir useopenssl
root@Youer:~/shiyan/shiyan01/shiyan1-1# cd useopenssl
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# openssl ecparam -genkey -
name SM2 -out sm2private_key.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# openssl ec -in
sm2private_key.pem -pubout -out sm2public_key.pem
read EC key
writing EC key
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# ls
sm2private_key.pem  sm2public_key.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# mv sm2private_key.pem
alice_private_key.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# mv  sm2public_key.pem
alice_public_key.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# ls
alice_private_key.pem  alice_public_key.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git commit -m "Alice
generated sm2 keys"
[master 7d5772f] Alice generated sm2 keys
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

2 files changed, 12 insertions(+)
create mode 100644 shiyan1-1/useopenssl/alice_private_key.pem
create mode 100644 shiyan1-1/useopenssl/alice_public_key.pem
```

- Alice将公钥发送给Bob，同时接收Bob的公钥

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# ls
alice_private_key.pem  alice_public_key.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# cp ./alice_public_key.pem /mnt/d/xlm20
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# cp /mnt/d/xlm20/bob_public_key.pem ./
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# ls
alice_private_key.pem  alice_public_key.pem  bob_public_key.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git commit -m "Exchanged public keys"
[master f669539] Exchanged public keys
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 4 insertions(+)
create mode 100755 shiyan1-1/useopenssl/bob_public_key.pem
```

- Alice发给Bob的明文plain.txt，内容为自己的姓名学号

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# echo "20221414xlm" > plain.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# ls
alice_private_key.pem  alice_public_key.pem  bob_public_key.pem  plain.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git commit -m "Created plain text file"
```

```
[master b24b22e] Created plain text file
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 shiyan1-1/useopenssl/plain.txt
```

- Alice：sm4 key使用gmssl rand 产生，16字节，记作k

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# openssl rand -hex 16 >
sm4_key.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# ls
alice_private_key.pem  alice_public_key.pem  bob_public_key.pem  plain.txt
sm4_key.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git commit -m "Generated
SM4 key"
[master 18aacc3] Generated SM4 key
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+)
create mode 100644 shiyan1-1/useopenssl/sm4_key.txt
```

- Alice：Sm4Enc(k,P) = C

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# openssl rand -hex 16 >
iv.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# openssl enc -e -sm4-cbc -
```

```
in plain.txt -out ciphertext.bin -K $(cat sm4_key.txt) -iv $(cat iv.txt)
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# ls
alice_private_key.pem  alice_public_key.pem  bob_public_key.pem
ciphertext.bin  iv.txt  plain.txt  sm4_key.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git commit -m "Encrypted
plain text with SM4"
[master 7f65908] Encrypted plain text with SM4
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

2 files changed, 2 insertions(+)
create mode 100644 shiyan1-1/useopenssl/ciphertext.bin
create mode 100644 shiyan1-1/useopenssl/iv.txt
```

- Alice：Sm2Enc(PKb,k) = KC

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# openssl pkeyutl -encrypt
-pubin -inkey bob_public_key.pem -in sm4_key.txt -out encrypted_key.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# ls
alice_private_key.pem  bob_public_key.pem  encrypted_key.bin  plain.txt
alice_public_key.pem   ciphertext.bin      iv.txt             sm4_key.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git commit -m "Encrypted
SM4 key with SM2 using Bob's public key"
[master 832e70b] Encrypted SM4 key with SM2 using Bob's public key
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 shiyan1-1/useopenssl/encrypted_key.bin
```

- Alice：Sm2Sign（SKa，C） = S1

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# openssl pkeyutl -sign -in
ciphertext.bin -inkey alice_private_key.pem -out signature.bin -rawin -
digest sm3
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# ls
alice_private_key.pem  bob_public_key.pem  encrypted_key.bin  plain.txt
sm4_key.txt
alice_public_key.pem   ciphertext.bin      iv.txt             signature.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git commit -m "Signed
ciphertext with SM2 using Alice's private key"
[master ed58865] Signed ciphertext with SM2 using Alice's private key
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 shiyan1-1/useopenssl/signature.bin
```

- Alice： 数字信封 C||KC||S1 发给Bob

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# cat ciphertext.bin
encrypted_key.bin signature.bin > digital_envelope.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# ls
alice_private_key.pem  bob_public_key.pem  digital_envelope.bin  iv.txt
signature.bin
alice_public_key.pem   ciphertext.bin      encrypted_key.bin     plain.txt
sm4_key.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# cp ./digital_envelope.bin
/mnt/d/xlm20
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git commit -m "Created
digital envelope"
[master 947b2c5] Created digital envelope
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
```

```
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 shiyan1-1/useopenssl/digital_envelope.bin
```

- git-log

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# git log
commit 947b2c5d27ba0291a796cb1dd4955050fb056038 (HEAD -> master)
Author: root <root@Youer>
Date:   Sun Oct 13 17:20:05 2024 +0800

    Created digital envelope

commit ed58865acdac3644e1b3126570cd1f0f4d7aa12d
Author: root <root@Youer>
Date:   Sun Oct 13 17:17:43 2024 +0800

    Signed ciphertext with SM2 using Alice's private key

commit 832e70b31947ad0741190acfc2813656292cb039
Author: root <root@Youer>
Date:   Sun Oct 13 17:10:38 2024 +0800

    Encrypted SM4 key with SM2 using Bob's public key

commit 7f65908176290c8af7c6136e18eeb47f7ffa8f44
Author: root <root@Youer>
Date:   Sun Oct 13 17:09:22 2024 +0800

    Encrypted plain text with SM4

commit 18aacc3ec98ad03b35aea5da07bdb9cb68592d35
Author: root <root@Youer>
Date:   Sun Oct 13 17:07:30 2024 +0800

    Generated SM4 key
```

- 一些问题：
  - openssl的sm2命令不直观，AI往往回答错误
    - 建议以老师的命令为蓝本
  - 在wsl中，如何实现与windows系统的文件的互通

4. 两人一组，在 Ubuntu或openEuler中（推荐 openEuler）中使用GmSSL命令实现带签名的数字信封协议。使用GmSSL，Bob发送，Alice接收。Ailice，Bob在实验中要替换为自己的8位学号+姓名。 使用Markdown记录详细记录实践过程，每完成一项git commit 一次。（10分）

- 生成自己的公私钥

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/useopenssl# cd ..
root@Youer:~/shiyan/shiyan01/shiyan1-1# mkdir usegmssl
root@Youer:~/shiyan/shiyan01/shiyan1-1# cd usegmssl
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# gmssl sm2keygen -pass
pass:5678 -out bob_sm2.pem -pubout bob_sm2pub.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# git commit -m "Generate SM2
key pairs"
[master 0e1f23c] Generate SM2 key pairs
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

2 files changed, 12 insertions(+)
create mode 100644 shiyan1-1/usegmssl/bob_sm2.pem
create mode 100644 shiyan1-1/usegmssl/bob_sm2pub.pem
```

- 与陆宇航交换公钥

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# ls
bob_sm2.pem  bob_sm2pub.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# cp ./bob_sm2pub.pem
/mnt/d/xlm20
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# cp
/mnt/d/xlm20/alice_sm2pub.pem ./
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# ls
alice_sm2pub.pem  bob_sm2.pem  bob_sm2pub.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# git commit -m "Exchanged
public keys"
[master d87b09d] Exchanged public keys
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
```

```
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 4 insertions(+)
create mode 100755 shiyan1-1/usegmssl/alice_sm2pub.pem
```

- 与曾庆林交换公钥和文件架构调整

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# ls
alice_sm2pub.pem  bob_sm2.pem  bob_sm2pub.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# mkdir zql_file
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# mkdir lyh_file
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# cd zql_file
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# cp
/mnt/d/xlm20/sm2pub.pem ./
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# ls
sm2pub.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# mv sm2pub.pem
alice_sm2pub.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# cd ..
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# cd lyh_file
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# cp
../alice_sm2pub.pem ./
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# ls
alice_sm2pub.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# cd ..
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# ls
alice_sm2pub.pem  bob_sm2.pem  bob_sm2pub.pem  lyh_file  zql_file
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# rm alice_sm2pub.pem
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# tree
.
├── bob_sm2.pem
├── bob_sm2pub.pem
├── lyh_file
│   └── alice_sm2pub.pem
└── zql_file
    └── alice_sm2pub.pem

2 directories, 4 files
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# git commit -m "Exchange of
public keys and file structure adjustment"
[master 367544e] Exchange of public keys and file structure adjustment
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
```

```
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

2 files changed, 4 insertions(+)
rename shiyan1-1/usegmssl/{ => lyh_file}/alice_sm2pub.pem (100%)
create mode 100755 shiyan1-1/usegmssl/zql_file/alice_sm2pub.pem
```

- 接受陆宇航的数字信封

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# cd lyh_file
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file#  cp
/mnt/d/xlm20/*.bin ./
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# cp
/mnt/d/xlm20/*.sig ./
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# ls
alice_sm2pub.pem  cipher.bin  iv.bin  key_encrypted.bin  sm2.sig
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# git commit -m
"Accept Lu Yuhang's digital envelope"
[master 2f72fb2] Accept Lu Yuhang's digital envelope
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

4 files changed, 2 insertions(+)
create mode 100755 shiyan1-1/usegmssl/lyh_file/cipher.bin
create mode 100755 shiyan1-1/usegmssl/lyh_file/iv.bin
create mode 100755 shiyan1-1/usegmssl/lyh_file/key_encrypted.bin
create mode 100755 shiyan1-1/usegmssl/lyh_file/sm2.sig
```

- 验证曾庆林的成果

```
```

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# cd zql_file
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# cp
/mnt/d/xlm20/*.bin ./
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# cp
/mnt/d/xlm20/*.cbc ./
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# ls
KC.bin  S1.bin  alice_sm2pub.pem  iv.bin  zqlsm4.cbc
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# gmssl sm2verify -
pubkey alice_sm2pub.pem -sig S1.bin -in zqlsm4.cbc
verify : success
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# rm KC.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# cp
/mnt/d/xlm20/KC.bin ./
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# ls
KC.bin  S1.bin  alice_sm2pub.pem  iv.bin  key.bin  zqlsm4.cbc
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# gmssl sm2decrypt -
key ../bob_sm2key.pem -pass pass:5678 -in KC.bin
gmssl sm2decrypt: open '../bob_sm2key.pem' failure : No such file or
directory
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# cd ..
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# ls
bob_sm2.pem  bob_sm2pub.pem  lyh_file  zql_file
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# cd zql_file
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# gmssl sm2decrypt -
key ../bob_sm2.pem -pass pass:5678 -in KC.bi
n -out key.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# ls
KC.bin  S1.bin  alice_sm2pub.pem  iv.bin  key.bin  zqlsm4.cbc
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# cat key.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# gmssl sm4_cbc -
decrypt -key $(xxd -p -c 32 key.bin) -iv $(xxd
-p -c 32 iv.bin) -in zqlsm4.cbc -out outcome.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# ls
KC.bin  S1.bin  alice_sm2pub.pem  iv.bin  key.bin  outcome.txt  zqlsm4.cbc
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# cat outcome.txt
20221418zqlroot@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# git add
.
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/zql_file# git commit -m
"finish zql task"
[master b4223f1] finish zql task
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

```

```
 6 files changed, 6 insertions(+)
 create mode 100755 shiyan1-1/usegmssl/zql_file/KC.bin
 create mode 100755 shiyan1-1/usegmssl/zql_file/S1.bin
 create mode 100755 shiyan1-1/usegmssl/zql_file/iv.bin
 create mode 100644 shiyan1-1/usegmssl/zql_file/key.bin
 create mode 100644 shiyan1-1/usegmssl/zql_file/outcome.txt
 create mode 100755 shiyan1-1/usegmssl/zql_file/zqlsm4.cbc
```

- 验证陆宇航的成果

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# ls
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# cp
/mnt/c/xlm20/*.pem ./
cp: cannot stat '/mnt/c/xlm20/*.pem': No such file or directory
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# cp
/mnt/d/xlm20/*.pem ./
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# cp
/mnt/d/xlm20/*.bin ./
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# gmssl sm2verify -
pubkey alice_sm2pub.pem -sig signature.bin -in encrypted_key.bin -id
20221425
gmssl sm2verify: open 'encrypted_key.bin' failure : No such file or
directory
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# tree
.
├── alice_sm2pub.pem
├── cipher.bin
├── iv.bin
├── key_encrypted.bin
└── signature.bin

0 directories, 5 files
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# gmssl sm2verify -
pubkey alice_sm2pub.pem -sig signature.bin -in cipher.bin -id 20221425
/root/GmSSL/src/sm2_sign.c:265:sm2_fast_verify():
/root/GmSSL/src/sm2_sign.c:671:sm2_verify_finish():
gmssl sm2verify: inner error
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# gmssl sm2verify -
pubkey alice_sm2pub.pem -sig signature.bin -in cipher.bin
verify : success
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# gmssl sm2decrypt -
key ../bob_sm2.pem -pass pass:5678 -in key_encrypted.bin -out key.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# gmssl sm4_cbc -
decrypt -key $(xxd -p -c 32 key.bin) -iv $(xxd
-p -c 32 iv.bin) -in cipher.bin -out outcome.txt

^C
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# ls
alice_sm2pub.pem  cipher.bin  iv.bin  key.bin  key_encrypted.bin
signature.bin
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# gmssl sm4_cbc -
decrypt -key $(xxd -p -c 32 key.bin) -iv $(xxd
-p -c 32 iv.bin) -in cipher.bin -out outcome.txt
^C
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# gmssl sm4_cbc -
decrypt -key $(xxd -p key.bin) -iv $(xxd
-p  iv.bin) -in cipher.bin -out outcome.txt
^C
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# KEY=$(xxd -p -l 16
key.bin)
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# IV=$(xxd -p -l 16
iv.bin)
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# gmssl sm4_cbc -
decrypt -key $KEY -iv $IV -in cipher.bin -out outcome.txt
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# cat outcome.txt
20221425lyh
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# git add .
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl/lyh_file# git commit -m
"Complete the verification work"
[master 917acb1] Complete the verification work
Committer: root <root@Youer>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

8 files changed, 6 insertions(+), 4 deletions(-)
create mode 100644 shiyan1-1/usegmssl/lyh_file/key.bin
create mode 100644 shiyan1-1/usegmssl/lyh_file/outcome.txt
create mode 100755 shiyan1-1/usegmssl/lyh_file/signature.bin
delete mode 100755 shiyan1-1/usegmssl/lyh_file/sm2.sig
```

- git-log:

```
root@Youer:~/shiyan/shiyan01/shiyan1-1/usegmssl# git log
commit 917acb1030c8fa4efbb95ddb85c7bf4669d7fec9 (HEAD -> master,
origin/master)
Author: root <root@Youer>
Date:   Sun Oct 13 21:27:24 2024 +0800

    Complete the verification work

commit b4223f186655b09a15c3869cbec9a46c98b9a03b
Author: root <root@Youer>
Date:   Sun Oct 13 19:36:32 2024 +0800

    finish zql task
```

```
commit 2f72fb291b6a4ddf682cbe36f627a1870202cd13
Author: root <root@Youer>
Date:   Sun Oct 13 19:01:38 2024 +0800

    Accept Lu Yuhang's digital envelope

commit 367544e1159de20d6d8d06265d1652c1c23ce925
Author: root <root@Youer>
Date:   Sun Oct 13 18:57:01 2024 +0800

    Exchange of public keys and file structure adjustment

commit d87b09d4935e7f7860a2898dc32f79b85a79bf16
Author: root <root@Youer>
Date:   Sun Oct 13 18:03:30 2024 +0800

    Exchanged public keys
```

---

5. 实验记录中提交 gitee 课程项目链接，提交本次实验相关 git log运行结果

- 实验一的Gittee链接
- git log运行结果：

```
root@Youer:~/shiyan/shiyan01# git log --oneline
917acb1 (HEAD -> master, origin/master) Complete the verification work
b4223f1 finish zql task
2f72fb2 Accept Lu Yuhang's digital envelope
367544e Exchange of public keys and file structure adjustment
d87b09d Exchanged public keys
0e1f23c Generate SM2 key pairs
947b2c5 Created digital envelope
ed58865 Signed ciphertext with SM2 using Alice's private key
832e70b Encrypted SM4 key with SM2 using Bob's public key
7f65908 Encrypted plain text with SM4
18aacc3 Generated SM4 key
b24b22e Created plain text file
f669539 Exchanged public keys
7d5772f Alice generated sm2 keys
230ab8b restart task3
5ed2b1a Encrypted plain text with SM4
9e17b9b Generated SM4 key
a495c41 Created plain text file
7199176 Exchanged public keys
ef827b4 Alice generated key pair
eab1cfa finish gmssl sm2 command
18446d1 finish gmssl sm4 command
8efe684 finish gmssl sm3 command
264f8f9 finish sm2 command
3bd6f65 finish RSA command
11df9cc finish enc command and create RSA keys
```

```
0f82610 finish asn1parse command
9c2859a finish base64 command
3dce8b6 Input and Output of Data in Different Bases
```

---

6. 提交要求：

- 提交实践过程Markdown和转化的PDF文件
- 代码，文档托管到gitee或github等，推荐 gitclone
- 记录实验过程中遇到的问题，解决过程，反思等内容，完成实验报告相关内容