Київський національний університет імені Тараса Шевченка
факультет радіофізики, електроніки та комп'ютерних систем

Лабораторна робота № 5

Тема: «Знайомство з бібліотекою OpenSSL»

Роботу виконав

студент 4 курсу КІ СА

Голубцов Владислав Романович

Київ 2021

Тема: Знайомство з бібліотекою OpenSSL

Мета: Ознайомитись з можливостями бібліотеки OpenSSL для шифрування та аутентифікації повідомлень. Отримання навичок з використання утиліти командного рядку openssl та API бібліотеки OpenSSL для створення власного коду, що використовує криптографічні функції.

Короткі відомості: OpenSSL - повноцінна криптографічний бібліотека з відкритим вихідним кодом, широко відома через розширення SSL / TLS, використовуваного в веб-протоколі HTTPS. Підтримує майже всі низькорівневі алгоритми хешування, шифрування і електронного підпису, а також реалізує більшість популярних криптографічних стандартів. OpenSSL доступна для більшості Unix-подібних операційних систем (включаючи Linux, macOS і BSD) і Microsoft Windows.

Хід роботи

1. Шифрування даних за допомогою OpenSSL

1. Оберіть файл над яким ви будете виконувати маніпуляції, наприклад /etc/passwd

Попрацюємо з паролями до 3 лабораторної роботи: passwords.txt

2. Ознайомтесь з загальним синтаксисом та ключами команди використовуючи вбудовану довідку: openssl enc help

```
C:\Program Files\OpenSSL-Win64\bin>openssl enc --help
Usage: enc [options]
Valid options are:
 -help               Display this summary
 -list               List ciphers
 -ciphers            Alias for -list
 -in infile          Input file
 -out outfile        Output file
 -pass val           Passphrase source
 -e                  Encrypt
 -d                  Decrypt
 -p                  Print the iv/key
 -P                  Print the iv/key and exit
 -v                  Verbose output
 -nopad              Disable standard block padding
 -salt               Use salt in the KDF (default)
 -nosalt             Do not use salt in the KDF
 -debug              Print debug info
 -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
 -bufsize val        Buffer size
 -k val              Passphrase
 -kfile infile       Read passphrase from file
 -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
 -rand val           Load the file(s) into the random number generator
 -writerand outfile  Write random data to the specified file
 -engine val         Use engine, possibly a hardware device
```

3. Виконайте шифрування файлу в режимі **AES-256-CBC** з використанням парольної фрази як джерела ключа та вектору ініціалізації (автоматична генерація за допомогою хеш-функції, зазвичай md5 чи sha1 за замовчуванням, в залежності від версії openssl):

```
OpenSSL>  enc -aes-256-cbc -in passwords.txt -out /passwd-aes-256-cbc -k supersecret
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

passwd-aes-256-cbc	11.11.2021 00:10	Файл	1 КБ

4. Порівняйте довжину оригінального та зашифрованого файлу.

```
18.10.2021  19:49               208 passwords.txt
```

```
11.11.2021  00:10               240 passwd-aes-256-cbc
```

Зашифрований файл збільшився на 32 байти.

5. Виконайте операцію дешифрування з правильною та неправильною парольною фразою:

```
C:\Program Files\OpenSSL-Win64\bin>openssl enc -aes-256-cbc -d -in passwd-aes-256-cbc -k supersecret
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
Mf3qU
hvC8w
mCux5
eOZYJ
2IbKu
5isdZ
rlBHC
27G1T
e6A1n
zcUQ5
92na1
Dbbpl
q9I01
oUTAG
lgrMy
EBqxK
skTRu
EqVee
E99GN
n3rtR
R40lM
r5v3B
qDuut
RTP3Z
x3Eiw
iL5pc
050rY
cmk3f
vjh3n
gTxNo
C:\Program Files\OpenSSL-Win64\bin>openssl enc -aes-256-cbc -d -in passwd-aes-256-cbc -k supersecret333
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
▯]C▯▯&Z▯
>H▯\9▯Åf/▯p▯▯▯PS▯Q}▯▯▯  G▯▯Ec▯gv3▯8j:9[(8▯ v▯

h8Zdlx▯L▯[$nA▯bad decrypt▯]:*:G#(_Cfa▯[W▯.▯N▯*oI
23256:error:06065064:digital envelope routines:EVP_DecryptFinal_ex:bad decrypt:crypto\evp\evp_enc.c:610:
```

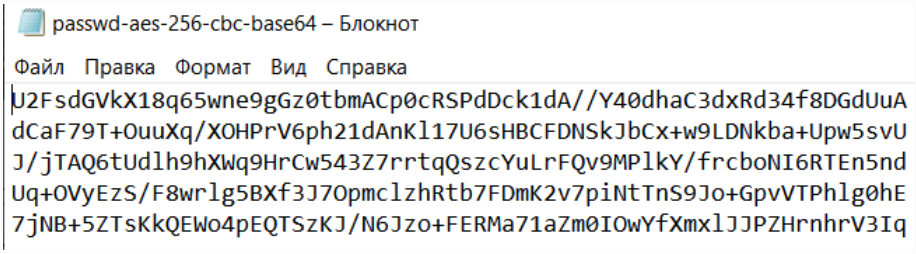
6. Для виводу шифротексту на екран та збереження в читаємому вигляді зазвичай використовується base64 кодування. Утиліта openssl містить вбудовані засоби роботи з base64:

openssl enc -aes-256-cbc -in /etc/passwd -base64 -out /tmp/passwd-aes-256-cbc-base64 -k supersecret

```
OpenSSL>  enc -aes-256-cbc -in passwords.txt -base64 -out passwd-aes-256-cbc-base64 -k supersecret
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

7. Перегляньте вміст base64-закодованого шифро-файлу.

```
C:\Program Files\OpenSSL-Win64\bin>type passwd-aes-256-cbc-base64
U2FsdGVkX18q65wne9gGz0tbmACp0cRSPdDck1dA//Y40dhaC3dxRd34f8DGdUuA
dCaF79T+OuuXq/XOHPrV6ph21dAnKl17U6sHBCFDNSkJbCx+w9LDNkba+Upw5svU
J/jTAQ6tUdlh9hXWq9HrCw543Z7rrtqQszcYuLrFQv9MPlkY/frcboNI6RTEn5nd
Uq+OVyEzS/F8wrlg5BXf3J7OpmclzhRtb7FDmK2v7piNtTnS9Jo+GpvVTPhlg0hE
7jNB+5ZTsKkQEWo4pEQTSzKJ/N6Jzo+FERMa71aZm0IOwYfXmxlJJPZHrnhrV3Iq
```



8. Порівняйте довжину оригінального та зашифрованого файлу. Дешифруйте в відкритий текст та переконайтеся в правильності результату.

```
11.11.2021  00:21               330 passwd-aes-256-cbc-base64
18.10.2021  19:49               208 passwords.txt
```

Файл збільшився на 122 байти

```
C:\Program Files\OpenSSL-Win64\bin>openssl enc -aes-256-cbc -d -a -in passwd-aes-256-cbc-base64 -k supersecret
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
Mf3qU
hvC8w
mCux5
eOZYJ
2IbKu
5isdZ
rlBHC
27G1T
e6A1n
zcUQ5
92na1
Dbbpl
q9I01
oUTAG
lgrMy
EBqxK
skTRu
EqVee
E99GN
n3rtR
R401M
r5v3B
qDuut
RTP3Z
x3Eiw
iL5pc
050rY
cmk3f
vjh3n
gTxNo
```

9. Для задання вектору ініціалізації та ключа явним чином (в шістнадцятковій системі) можна використати відповідні параметри (*спробуйте використати інші шифри та режими для різномаїття*):

```
OpenSSL> enc -des-ofb -in passwords.txt -out passwd-des-ofb-kiv -K FF02 -iv 0
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
```

```
OpenSSL> enc -des3 -in passwords.txt -out passwd-des3 -K FF02 -iv 0
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
```

```
OpenSSL>  enc -aes-256-cfb -in passwords.txt -out passwd-aes-256-cfb -k supersecret
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
```

10. Дешифруйте наступні шифротексти (використовується md5 для генерації ключа та вектора ініціалізації):

1.

```
U2FsdGVkX1+u3UiVivoMk7BiC8i/fNffF/aFA9+cqSmXPQeNmuhOgGYp70AGfwb8
xAFSiwSx/QtQ8YPHNcmSO/TmhSmc4AVXu9WHoSOiPIUXLlwrlG46ZK2+AcoLp3MZ
CjVkdLz33fnm0MYgp88bHkorO6MXOnoDSs3jitmv6i4Iul/3JqSMKk4u5QC9g0XO
QZFwSD8TEzbhdW+jpYI9al9dTf7U6ItAxgVYWjcnJIVgMN7ChyY1BLIHx4mXnXbk
fMq8/Rxa4/t3mj8G515dBRAIkYoMarFvN3iPEQDpHpKy2WiZhVrHXg==
```

HMAC-SHA256(plaintext) = b9d016f2909ea26d3aff1d5e081fb99b06b45929281cf134d634aaed0e44ff74
Парольна фраза: `cryptography`
Шифр: des-cbc
Ключ HMAC: hmackey

```
OpenSSL> enc -des-cbc -d  -base64 -in 1 -k cryptography -md md5
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
Она: ответь мне, только честно, да или нет, хорошо?
Он: спрашивай
Она: почему мужчины смеются над блондинками?
Он: да
```

2.

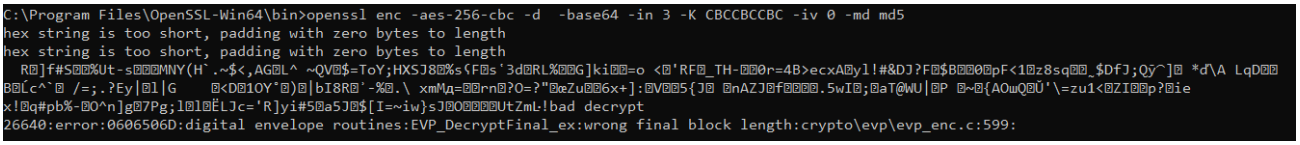
wib3KS9O6V+Zv9HC0xbDD27/KVkyhAVhXwz8rQc6DYqANM1Aj+LVHeKOtMoAEdC5
mRnUZF+64KPSamP2XiPWfXH+sLyVScqaxhECWect5LTZZqqFyb26I5gu9o68+c6B
w/rg+EZRvSWW+goMJozsMqxUqiBPhfX5d8iQwV7b1VnOXzW7wZjdhybV5jb5rMqu
1XqzWSbU7+kGKOPihXJlPZfO+Nb82eewZH/S9wZywE9PWAC9kpMUDCBChnbv3kcX
rlQEOwNIXXMwVrn8RmiJ0A62MbcmW8XIRSKHFujcPeJtrFiZ1FRH+/D6YfOUmJxp
dAaG4RmFlzl9EInqwdH3QYNPL5i2dItocgZntvW1Q3RF+Q+T0Dwzf5pO7eB/k6oo
a0zbmYJW3oAvs4/45ZunzMKDcpmGxaWlzg1k9dzka9a2AA8M8Z4NCa1DP4kXsoFT
TKCFNJRaa4wUtdg=

HMAC-SHA1(ciphertext) = 8a7fdd38030b04ca9135825b85ae45ff2821adc3
Ключ: BEBEBE
Вектор ініціалізації: 12345
Шифр: aes-128-ofb

```
OpenSSL> enc -aes-128-ofb -d  -base64 -in 2 -K BEBEBE -iv 12345 -md md5
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
У Львові на футболі фани заводять «Хто не скаче, той москаль!». Підривається півстадіону. Позаду сім'я (тато, мама, дитина років 5ти).
Дитина питає:
— Пап-пап, а што ані крічат?
— Пригай давай...
```

DmZ+nmwZEzWhdBiZby38y6O6rz5XrmljxRv72OSo+DEbk5r81PWkE9CJKDDfAFf6NruQ7ozeUcUi
Z86EKP6YXeFhpE0f3LFfi85FxtJU0BLYGW/86hbDgXCowXvi7RmrOypZjcy37pFB7mYEFE8+CsRO
palbzTPJMhN0FUdTt6dwiWFxe09TuBYeCjLSN++X+agInMgdjsaLauJyA9AauhYksExmB3qwkTOd
mbJgQWHKXT1hDLLDbVQc19dDDGpVMn7jPsEZMqCZbaGOK3qAnsRWBFvD1nhGPFl9dH+M39OWeD3J
sZo2dK5oCDMWh6zV5ZfKhz4+CgjeQ4J48+y8ScLPynK2PYU0ZDHYIC4ImF2x0roXsh3xDtsNR3nw
aQDomckMGLL6Xpq2IQn/2xO9Q/RWYKzDuyOamFHM5sGR7ok3kwaoK/eiWsxj5g083KoJgnlUT3PJ
aHksDfyF+tjdvmhm0BVfYZy56QOJeaQn15TvWIatqJ2lGCt7QysrwrWzvn2bNWLQW22eOBE1QWtF
HSEF9EFkiMbrb5jx8te+Kxiys0xAqCN9trm0t1Oc5e7bnvQadNRjYaMB17fhxi9Crv+qh7yvkP9A
5tQKF1x+ho78Ae1+iWmIzD55MeqjGNdQrYOL9n9KNsrMzfQqxUAHekOMQx3RItdQZOXTvegr+G08
bG/APQh5Bhs57E4xZSiR1HbroXaesdKwXYu9coLnx0WnR2QLw6u6p1GrmpjEpbH6BfvwBj4/bxD4
0wVkOQoIzsCb7OUu/zqZUH+9ZSqDH+C9F3LHuSL/oUuWKRGSBInHKxRdLljl2z14WB34EJ78NKha
K8zRJ1pABz5LBmJpUv2p59AQ40JIB+aEqpeCYlXq9Vfn4drEpUUniUATA0xWyiTApL/g9Md8loc7
vh8bESMiXsRGfveKSVjRKP5it2QRE00sneluFBNMgICisDGFvcbJAhovEg==

HMAC-MD5(base64 encoded ciphertext) = 916032deeaaa720523435f47121e26db
Ключ: CBCCBCCBC
Вектор ініціалізації: 0
Шифр: aes-256-cbc
Ключ HMAC: hmackey



На великий жаль третій шифротекст не вдалося розшифрувати, через помилку в шифруванні.

2. Аутентифікація повідомлень за допомогою OpenSSL

Для аутентифікації використовується команда openssl dgst. Команда дозволяє як порахувати значення хеш-функції чи HMAC так і використовувати асиметричну криптографію для генерації коду аутентичності.

Самостійно ознайомтеся з синтаксисом команди саме в аспекті генерації симетричних HMAC кодів. Для заданих в попередньому завданні текстів перевірте (відтворіть команду, що згенерує) значення кодів аутентичності. Наведіть команди в звіті.

Створимо розшифровані шифрофайли з пункту 1:

```
C:\Program Files\OpenSSL-Win64\bin>openssl enc -des-cbc -d -base64 -in 1 -out 1decrypted -k cryptography -md md5
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.

C:\Program Files\OpenSSL-Win64\bin>openssl enc -aes-128-ofb -d -base64 -in 2 -out 2decrypted -K BEBEBE -iv 12345 -md md5
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length

C:\Program Files\OpenSSL-Win64\bin>openssl enc -aes-256-cbc -d -base64 -in 3 -out 3decrypted -K CBCCBCCBC -iv 0 -md md5
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
bad decrypt
16524:error:0606506D:digital envelope routines:EVP_DecryptFinal_ex:wrong final block length:crypto\evp\evp_enc.c:599:
```

Через помилку в третьому, виконаємо дії з першим та другим текстом

HMAC-SHA256(plaintext) = b9d016f2909ea26d3aff1d5e081fb99b06b45929281cf134d634aaed0e44ff74

```
C:\Program Files\OpenSSL-Win64\bin>type 1decrypted |openssl dgst -sha256 -hmac "hmackey"
(stdin)= b9d016f2909ea26d3aff1d5e081fb99b06b45929281cf134d634aaed0e44ff74
```

HMAC-SHA1(ciphertext) = 8a7fdd38030b04ca9135825b85ae45ff2821adc3

```
C:\Program Files\OpenSSL-Win64\bin>type 2decrypted | openssl dgst -sha1 -hmac "hmackey"
(stdin)= 170fa8f67ca96fa45eef6cdc35c8655a975c1c29
```

3. Використання API бібліотеки OpenSSL для шифрування даних

Серед функцій утиліти openssl режими роботи блочного шифру приймають на вхід лише значення ключа в шістнадцятковому представленні або генерують його з використанням хеш-функції. Для використання будь-якого довільного ключа, можна досить просто реалізувати програму, що використовує готові функції API бібліотеки.

Ваше завдання - створити програму, що виконує шифрування (та дешифрування) за допомогою режиму **AES-128-CTR** довільного тектсу. Ключ та вектор ініціалізації задаються як аргументи командного рядку і є довільним текстом довжиною 128 біт (ASCII кодування).

Загальні відомості про API та приклади послідовності його використання можна знайти на вікі проекту: https://wiki.openssl.org. Почніть ознайомлення з загальної інформації про EVP-інтерфейс , ознайомтесь з процесом Symmetric Encryption and Decryption, зокрема підрозділ Encrypting the message з прикладом коду мовою C). Цільова функція для шифрування EVP_CIPHER *EVP_aes_128_ctr(void);

Нативним чином бібліотека орієнтована на мову C, проте існують зв’язки(binding) та обгортки(wrappers) під більшості інших мов, наприклад:

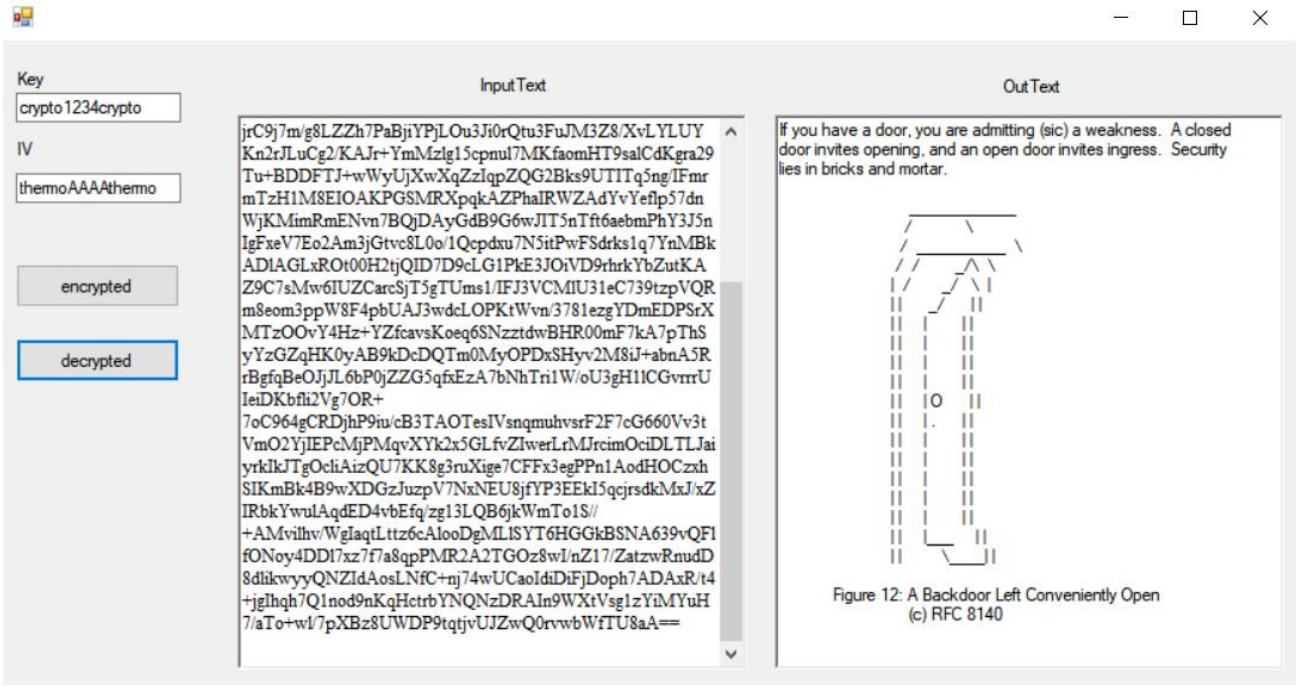
- .NET - https://github.com/openssl-net/openssl-net(абовикористанняMicrosoft’sSSPI)
- Python - https://github.com/pyca/pyopenssl
- для Java стабільної реалізації не існує, використовують зазвичай BouncyCastle

Для перевірки коректності роботи, дешифруйте наступний фрагмент base64-encoded шифротексту:

7IwDEpdhsP41iKPnRGEWe1HUMASC3nPY8OGvfwCBmeG1rK2drdhEQG1nK0BHcQVXXpQZ/iICZ51My1tvB6BEcEewJaArY
1K2eVsS1KwIdap9YZrPlKRKZloW/jjgJTGD5v3FPANDlXygiMSWdtLNTDwm1pQ9an3ko3q21p2PSAD3B2f/++KfYhjxcoAjqFM
hsbINhpbLtmTMjFQgB45T+62VnSE1yYprP/1iFd7ejfUg+wtZzkT+Dxl1joL198x+QmLHzSxh2ZiXlA1kwJcyjoQwZVIKAUC47H1
nvR2ZZN0tPhA5LwU0YxQENHcN/3aERi7ayFRmHweCE9YWIALeHjTGg80MOzo+6+vM7jgYWjilBGBpT/KNRgBldX0XYy31
e3nA8lHNmgN/D23zbTV1TF1oV2L0FL1FpjmfDJtNbVuheZJ2MlnnhcP/BlO2rwrmGjrC9j7m/g8LZZh7PaBjiYPjLOu3Ji0rQtu3FuJ
M3Z8/XvLYLUYKn2rJLuCg2/KAJr+YmMzlg15cpnul7MKfaomHT9salCdKgra29Tu+BDDFTJ+wWyUjXwXqZzIqpZQG2Bks9UT
ITq5ng/IFmrmTzH1M8EIOAKPGSMRXpqkAZPhaIRWZAdYvYeflp57dnWjKMimRmENvn7BQjDAyGdB9G6wJIT5nTft6aebmP
hY3J5nIgFxeV7Eo2Am3jGtvc8L0o/1Qcpdxu7N5itPwFSdrks1q7YnMBkADlAGLxROt00H2tjQID7D9cLG1PkE3JOiVD9rhrkYbZu
tKAZ9C7sMw6IUZCarcSjT5gTUms1/IFJ3VCMlU31eC739tzpVQRm8eom3ppW8F4pbUAJ3wdcLOPKtWvn/3781ezgYDmEDPSr
XMTzOOvY4Hz+YZfcavsKoeq6SNzztdwBHR00mF7kA7pThSyYzGZqHK0yAB9kDcDQTm0MyOPDxSHyv2M8iJ+abnA5RrBgf
qBeOJjJL6bP0jZZG5qfxEzA7bNhTri1W/oU3gH1lCGvrrrUIeiDKbfli2Vg7OR+7oC964gCRDjhP9iu/cB3TAOTesIVsnqmuhvsrF2F7
cG660Vv3tVmO2YjIEPcMjPMqvXYk2x5GLfvZIwerLrMJrcimOciDLTLJaiyrkIkJTgOcliAizQU7KK8g3ruXige7CFFx3egPPn1Aod
HOCzxhSIKmBk4B9wXDGzJuzpV7NxNEU8jfYP3EEkI5qcjrsdkMxJ/xZIRbkYwulAqdED4vbEfq/zg13LQB6jkWmTo1S//+AMvil
hv/WgIaqtLttz6cAlooDgMLlSYT6HGGkBSNA639vQFlfONoy4DDl7xz7f7a8qpPMR2A2TGOz8wI/nZ17/ZatzwRnudD8dlikwyyQ
NZIdAosLNfC+nj74wUCaoIdiDiFjDoph7ADAxR/t4+jgIhqh7Q1nod9nKqHctrbYNQNzDRAIn9WXtVsg1zYiMYuH7/aTo+wl/7pX
Bz8UWDP9tqtjvUJZwQ0rvwbWfTU8aA==

Ключ: crypto1234crypto (ASCII)
Вектор ініціалізації: thermoAAAAthermo (ASCII)

Результат:



Посилання: https://github.com/VladyslavHolubtsov/ZIKS

Висновки

У ході лабораторної роботи я ознайомився з можливостями бібліотеки OpenSSL для шифрування\дешифрування та аутентифікації повідомлень. Отримав навички з використання утиліт командного рядку openssl та API бібліотеки OpenSSL для створення власного коду, що використовує криптографічні функції.