

Міністерство освіти і науки України  
Національний технічний університет «ХПІ»  
Навчально-науковий інститут комп'ютерних наук та інформаційних  
технологій  
Кафедра комп'ютерної інженерії та програмування

## **ЗВІТ**

з лабораторної роботи № 4  
з дисципліни «Сучасні технології безпечного програмування»  
**«ВИКОРИСТАННЯ МНЕМОНІЧНИХ ФРАЗ ДЛЯ ФОРМУВАННЯ  
КЛЮЧІВ ШИФРУВАННЯ»**

Виконав:  
студент гр. КН-Н9226  
Кулик Д.І.

Перевірів:  
Бульба С. С.

**Мета роботи:** Дослідити і реалізувати механізм використання мнемонічних фраз для формування ключів шифрування

### **Індивідуальне завдання**

- Використовуючи алгоритм `bir39`, створити `seed` генератора псевдовипадкових чисел за допомогою мнемонічної фрази та стосовні ключі шифрування.
- Зашифрувати текст
- Використовуючи раніше створену мнемонічну фразу, відновити ключі шифрування на дешифрувати текст. Вдосконалитись, що оригінальний та дешифрований тексти однакові.

З.І., для додаткових балів необхідно реалізувати підтримку україномовних мнемонічних фраз.

### **Хід роботи**

VIP описує реалізацію мнемонічного коду або мнемонічного речення — групи слів, які легко запам'ятати — для створення детермінованих гаманців. Він складається з двох частин: створення мнемоніки та перетворення її на бінарне початкове значення. Пізніше це початкове число можна використовувати для створення детермінованих гаманців за допомогою VIP-0032 або подібних методів.

## Важливі фрагменти програми

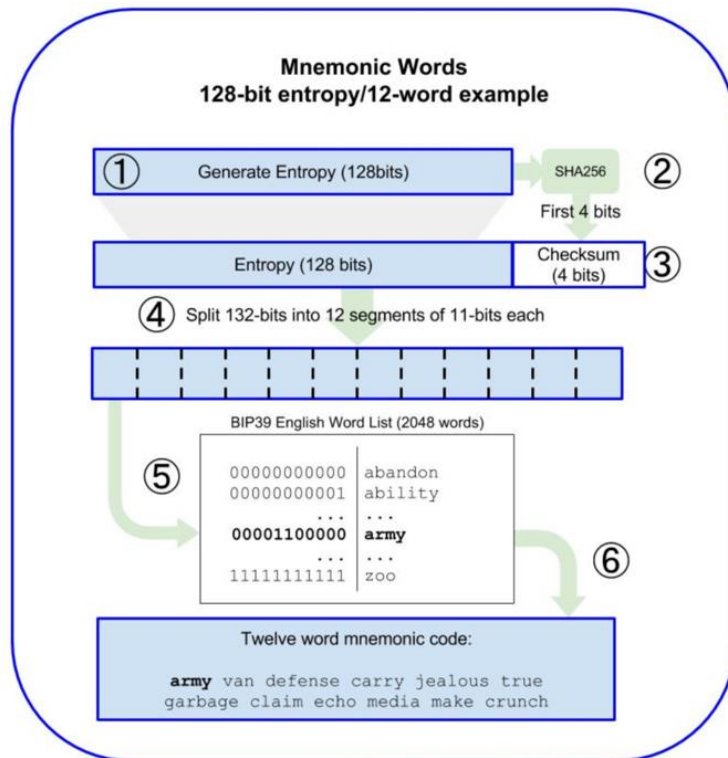


Рисунок 1 – Алгоритм створення ключів

```
def get_2048_words(lang):  
    if lang == 'en':  
        df = pd.read_csv(  
            './kulyk04/en_words.txt',  
            names=['words']  
        ).reset_index().set_index('words')  
  
        nums = df.to_dict()['index']  
    elif lang == 'ua':  
        df = pd.read_csv(  
            './kulyk04/ua_words.txt',  
            names=['words']  
        ).reset_index().set_index('words')  
  
        nums = df.to_dict()['index']  
  
    return nums
```

Рисунок 1 – Зчитування слів зі словників

```
def get_seed(mnemonic: str, passphrase: str = "") -> bytes:
    decode_phrase(mnemonic)
    mnemonic = normalize_string(mnemonic)
    passphrase = "mnemonic" + normalize_string(passphrase)
    mnemonic_bytes = mnemonic.encode("utf-8")
    passphrase_bytes = passphrase.encode("utf-8")
    stretched = hashlib.pbkdf2_hmac(
        "sha512", mnemonic_bytes, passphrase_bytes, PBKDF2_ROUNDS
    )
    print(f'GENERATED SEED      : {stretched.hex()}')
    return stretched
```

**Рисунок 3 – Генерація чисел**

```
def encrypt(text, mnemonic):
    rsa = RSA.generate(1024, randfunc=PRNG(get_seed(mnemonic)))
    public_key = rsa.public_key().export_key('PEM')
    print(f'PUBLIC KEY          : {public_key}')
    cipher = PKCS1_OAEP.new(RSA.import_key(public_key))
    encrypted_text = cipher.encrypt(text)

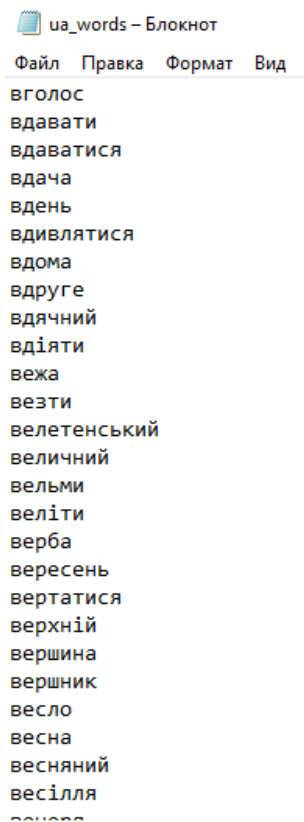
    return encrypted_text
```

**Рисунок 4 – Функція шифрування**

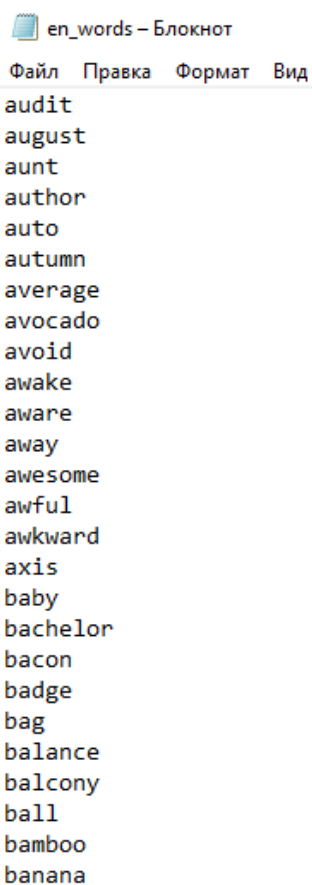
```
def decrypt(encrypted_text, mnemonic):
    rsa = RSA.generate(1024, randfunc=PRNG(get_seed(mnemonic)))
    private_key = rsa.export_key('PEM')
    print(f'PRIVATE KEY           : {private_key}')
    cipher = PKCS1_OAEP.new(RSA.import_key(private_key))
    decrypted_text = cipher.decrypt(encrypted_text)

    return decrypted_text
```

**Рисунок 5 – Функція дешифрування**



**Рисунок 6 – Словник українських слів**



**Рисунок 7 – Словник англійських слів**

## Результати роботи програми

```
(stbp_venv) C:\tmp\stbp>python kulyk04\main.py
ORIGINAL TEXT      : b'Daniil Kulyk'
Mnemonic           : каміння каміння каміння каміння каміння каміння каміння каміння каміння каміння мисливець

----- ENCRYPTION -----

GENERATED SEED      : abedc45093e798f8f04e4109b79be098e428f5aeb12b754e144b5a7b607560f7a5139c8689cab51d6febd33d660ec6d
ec1c8d412214c552fbd9b202b2411ce
PUBLIC KEY          : b'-----BEGIN PUBLIC KEY-----\nMIGfMA0GCsGqSIb3DQEBAQUAA4GNADCBiQKBgQC1UtYntV1arhw1x8Lx8wVeUk2h\ne
aIyhqXBe/1ayR6T1Hx6BH1gs8mWCaGmoE15z/URfGnJs+ZG+EX1UjM1044SzREb\ndFz9hbpWdnKL9CwMMWB5YM+gYQ7+j9I7tz0GoLOuMXPDB8iuqppSng0
d0ewhplF0\XnVeyR8z/cxT69mQNOIDAQAB\n-----END PUBLIC KEY-----'
ENCRYPTEd TEXT     : 1adeffa601c96ac43353847c72c934907b2b4f3c1daf6a90daf55cb22b48f978a25b5020a081d1a4d60982b58a69d450e
25f706634127d17dcf6bf6ff79199fb808737ec9230f149df39a748aa78336ee3d97a8baa757ba5a45579134afe6e10f28ab3e048582450e686c426f
8fe7b283e1f83020e345f35dfa4ff2c826775fa

----- DECRYPTION -----

GENERATED SEED      : abedc45093e798f8f04e4109b79be098e428f5aeb12b754e144b5a7b607560f7a5139c8689cab51d6febd33d660ec6d
ec1c8d412214c552fbd9b202b2411ce
PRIVATE KEY         : b'-----BEGIN RSA PRIVATE KEY-----\nMIICXAIBAAKBgQC1UtYntV1arhw1x8Lx8wVeUk2heaIyhqXBe/1ayR6T1Hx6BH
lgVns8mWCaGmoE15z/URfGnJs+ZG+EX1UjM1044SzREbDfZ9hbpWdnKL9CwMMWB5YM+gY\nQ7+j9I7tz0GoLOuMXPDB8iuqppSng0d0ewhplF0\XnVeyR8z/c
xT69mQNOIDAQAB\nAoGAGHm5/gmsuloca215LtmXBAi3GZB+XMchhj/A521fHXKfFKDw/DG1gE3kl8u\nnFQzf5K70BZINiPniD0rFW2vfj72Yny6UDAAJLR
mEU02HvqLRTZ9us33djbjVV7Lb\nlPnjduT4MsXj9xoZZqf4mRg7RPh424Vs/1eL2VL9A/snWycCQQR5JKeEIMF/a5b\ncZQutyv4Zw7cFhkr9FYKCLbc
kHwWgaVZagu1In7bBR+v9F4pj2ZIk54KZF+5F\nnRZ8ARRADAKEA3cyB077F41IJOiX0yxadzyK7VSF/w/vgww4raca/5CPR6ZgLe1A\nnTR4TVLzn381Bwd
8hgC18F3e31ohIc+q1ZwJBANAel09qQY2vm/frlqpXE6bCKNuc\nuU7ZITFUQ7hjkV02PvSRMF1eanCyeajMxZY1pJ9BkwUvVao6ZWRinCmPskCQGTX\nnPr
AQaqpWm27GwfH0TC/qj/296z9+kda14u/FLTcXEs7vM5bVTA9PvPcnKFWL93I\nnw+jjmwls/b6IpC/dRj8CQDLZGw16HxWrZSY1MUNOTouJiHf4FT+Cck/n
g03ONXQ6\nnqoKL4MWTJ5ZetWfbvneNU9WB5a0a6K25c3Cpds5GTAA=\n-----END RSA PRIVATE KEY-----'
DECRYPTEd TEXT     : b'Daniil Kulyk'

(stbp_venv) C:\tmp\stbp>
```

Рисунок 8– Результат виконання програми для иа мнемонічних фраз

```
(stbp_venv) C:\tmp\stbp>python kulyk04\main.py
ORIGINAL TEXT      : b'Daniil Kulyk'
Mnemonic           : admit achieve appear awesome behind border bullet casino admit clean admit abstract

----- ENCRYPTION -----

GENERATED SEED      : e7d692a0ad33b61d31a7cd5f28d871d418c42981e7ca540abaa4964ab39f273e2a20d74fd5afc7e965344ad46c339ea
ce819612468cf29938caf2f9b2d6862
PUBLIC KEY          : b'-----BEGIN PUBLIC KEY-----\nMIGfMA0GCsGqSIb3DQEBAQUAA4GNADCBiQKBgQCX3A/golX/FfttWUcUpY1LFabU\nnQ
UdgZvm84DT6wsGcrVJ1VyKW6Mo1rvmy2KuAcfKIjEizrCAEKcd5licLgEcUvj4\nlu4kXFHXLPrMiK1FPjP1R54w6VyuYBA1SjtwXDerqzLT8n80doyEcX
8AP0jKFZ1\n2pbk6yeKgPueoCrvFQIDAQAB\n-----END PUBLIC KEY-----'
ENCRYPTEd TEXT     : 8da5024d18bce5705483d66925ab7bb4997e5b8846121624ffe89cfa117fe0ea3e7b0f4944cdc37fb11e34a0acf3d32
d53879ab28167a805b4bbbf7fa454dd18b4cd5e9b361da7fa7494d334efc171940a3bd0cb00739b94468feb995be4efeace52d281b9464b7477bf137
241a5e0830f77143c8c40c11f56c01089e5ae38

----- DECRYPTION -----

GENERATED SEED      : e7d692a0ad33b61d31a7cd5f28d871d418c42981e7ca540abaa4964ab39f273e2a20d74fd5afc7e965344ad46c339ea
ce819612468cf29938caf2f9b2d6862
PRIVATE KEY         : b'-----BEGIN RSA PRIVATE KEY-----\nMIICWQIBAAKBgQCX3A/golX/FfttWUcUpY1LFabUQUdgZvm84DT6wsGcrVJ1Vy
KW\n6Mo1rvmy2KuAcfKIjEizrCAEKcd5licLgEcUvj4lu4kXFHXLPrMiK1FPjP1R54w\n6VyuYBA1SjtwXDerqzLT8n80doyEcX8AP0jKFZ12pbk6yeKgP
ueoCrvFQIDAQAB\nAn8ByouD8EQIoYctDCJ9UtmczUwBWpZgZLgc5gt/j2dEjZcfF0tK8wJvhi7LKZ4\nn87BPPPV6YPkArUuMt8bcmfKEpY76I/kzmyh/6j
M9ilgxPxjSxSv4uHc+/ckdYWNQ\nnuF+/E6w/RbcGhBpzKu1ufE3708KztFuAJ1/9PxnT0BBZAKEatq6MQbOPEK0YD0BU\nnN/6pLFaw4AEBBuNhhkCH5FV76PZ
6Za0c7GYFvTDWUYs10jJb9DycFotFfC1kg2wrh\nne+KEyQJBANTot/SoXEJttMqPA51LVXutCSCuqjuQ9F4JqokkcrVMac5UaYN/Dvi\nn02VU63t1Y23JUQ
rqVP88ij3ZwDgbee0CQGcNVJgNHRpFQie378tAj41drFJYdYB8\ngT46lnmTH10eFZw7o6LRZE+1Kd4m1LLygcaw0YovsEypory70YY7PECQEFjPeyo\nnLt
dJ4t8h93j57PdXvv3jbhHkki4rdf+XS2+swLNEpsJ5nUkmpBsm1nk97piufORN\nnwHaJmajbt/YEGNECQC7mLsTryWlj+VBDtET5ZDRcS0ek13P4+QK1NOvA
wnRhYNWj\nncUwgXGr628h0j2GRudgTVrGyIQQGPdTL0iPELQ=\n-----END RSA PRIVATE KEY-----'
DECRYPTEd TEXT     : b'Daniil Kulyk'

(stbp_venv) C:\tmp\stbp>_
```

Рисунок 9 – Результат виконання програми для еп мнемонічних фраз

**Висновки:** в результаті виконання лабораторної роботи було досліджено і реалізовано механізм використання мнемонічних фраз для формування ключів шифрування RSA.