

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
УКРАЇНИ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ
СІКОРСЬКОГО»
Фізико-технічний інститут

Криптографія
Комп'ютерний практикум №4

Виконали:
Студенти групи ФБ-05
Сапожник М.В.
Карась Б.І.

Київ - 2022

Мета та основні завдання роботи:

Ознайомлення з тестами перевірки чисел на простоту і методами генерації ключів для асиметричної криптосистеми типу RSA;

практичне ознайомлення з системою захисту інформації на основі криптосхеми RSA, організація з використанням цієї системи засекреченого зв'язку й електронного підпису, вивчення протоколу розсилання ключів.

Хід роботи:

В рамках виконання практикуму було створено функції:

sign - функція для підпису повідомлення

SiCh - перевірка надісланого підпису

Encrypt- зашифровує повідомлення

Decrypt-розшифровує повідомлення

rsa_pair - генеруватиме d,n,e

TestPrime – тестує початкове число на простоту

g_c_d – розширений алгоритм Евкліда

f_unc –перевірка шляхом теста Міллера


Вивід виконання програми:

```
p1 = 48933315762255109757786971470274400141719751249010777590127842715250502709283
q1 -> 18580488624295996445122707961769826518061032903932507817494238869806561189863
p2 -> 75736591966363531269791265365900739838849364044243555942273995889417116441571
q2 -> 83063957224229514074146661983289508064204241151634137043709970799477180244563

d1 -> 204004734769190296422485514767101845095544038260250216434451835090727984829021080173118955533863648069410631122274568989246247926697584544586422573827533
d2 -> 1063391487407019316627946658336120919190793317008276347527761377763309949068569025192974162221513786404427850970974094217970905043966720936290640760345433
e -> 65537
n1 -> 909204916869665042954126703930061449984812623968855384866689555684531787945633955919700865205000726988168156559914349151934742018626227882235201655598229
n2 -> 6290981035402945022011711513574143047572397690627496568688111339093161593438216952207607967075976635127835565864867236017468779287718516012336098079928473
M -> 60470707704955905611759442832632685357434441451689967754614772847051907307440
C -> 15712171613497240436733468674798887146160360492093533399562698294597409157559256480693033429647824737756905664251203805481309867413699244594269826716786
S -> 2154174517316172824035580561097098920233590728701332071061009177274178945494422094961447836164606780095019342552329917187461791868342174771435546242354572
M' -> 60470707704955905611759442832632685357434441451689967754614772847051907307440
```

Починаємо перевірку за сайтом

Get server key

 Clear

Key size

256

Get key

Modulus

B820DBA73C3FD23181372C74F18AE5FF883008DC01B9B47906C17F436D3340CA5

Public exponent

10001

Encryption

✖ Clear

Modulus

B820DBA73C3FD23181372C74F18AE5FF883008DC01B9B47906C17F436D340CA5

Public exponent

10001

Message

65537

Bytes ▼

Encrypt

Ciphertext

96D5D9EB1D47902500438EB41C67E499AC8CEF2C5C248623E3F36FB995FE7861

Sign

✖ Clear

Message

65537

Bytes ▼

Sign

Signature

67857205CEAF5A3549C7E5747B8494B792D2DDC1341856997784BD492B875402

Verify

✖ Clear

Message

65537

Bytes ▼

Signature

67857205CEAF5A3549C7E5747B8494B792D2DDC1341856997784BD492B875402

Modulus

B820DBA73C3FD23181372C74F18AE5FF883008DC01B9B47906C17F436D340CA5

Public exponent

10001

Verify

Verification

true

✓

Переробимо деякі моменти у коді для перевірки та додамо деякі значення:

```

class SiCh:
    def __init__(self, m, s, e, n):
        self.m = m
        self.s = s
        self.e = e
        self.n = n

    def sich(self):
        if self.m == pow(self.s, self.e, self.n):
            print("Vdaloysa!")
        else:
            print("Nevdacha!")
        return self.m == pow(self.s, self.e, self.n)

n = int("B820DBA73C3FD23181372C74F18AE5FF883008DC01B9B47906C17F436D340CA5", 16)
print(f'n: {n}')
e = int("10001", 16)
print(f'e: {e}')
msg = int("65537", 16)
print(f'Message: {msg}')
encrypted_msg = Encrypt(msg, e, n).enc_msg()
print(f"Ciphertext: {hex(encrypted_msg)}")
sign = int("67857205CEAF5A3549C7E5747B8494B792D2DDC1341856997784BD492B875402", 16)
print(f"Sign is: {sign}")
SiCh(msg, sign, e, n).sich()

```

```

n: 83283619236499956271491479917924327265085160990819505273662546670712421354661
e: 65537
Message: 415031
Ciphertext: 0x96d5d9eb1d47902500438eb41c67e499ac8cef2c5c248623e3f36fb995fe7861
Sign is: 46824001019341698445336993165757948483140514894172248168801032960957624046594
Vdaloysa!

```

Висновки

В результаті виконання практикуму ми практично ознайомились з системою захисту інформації на основі криптосхеми RSA. Було засвоєно і практично використано тест Міллера-Рабіна для перевірки чисел на простоту і генерації простих чисел. Мали змогу використати Asym Crypto Lab Environment і з його допомогою перевірити нами створені функції криптосхеми RSA.