Лабораторная работа 4
Атака на алгоритм шифрования RSA, основанная на Китайской теореме об остатках
Вариант 1

Выполнила: Батомункуева Виктория

Группа: P34101

Цель работы:

Изучить атаку на алгоритм шифрования RSA посредством Китайской теоремы об остатках

Вариант:

Модули, N1 – 359690807803, **N2** – 361062169537, **N3** – 363514381513
Блок зашифрованного текста:
C1 – 177412278620 8631904062 60910035474 297496979396 44306701511 223949114264 95163574676 126740768642 306466049596 82343556476 97754924718 242675823829
C2 – 227891126441 175684889961 108275398403 50799922679 50861774819 120598551775 214220319631 193858968963 243446962166 168236630688 260389624172 86845867002
C3 – 230974691188 345734293737 118726556071 220369632983 69236028918 121704957571 269179568504 201685371953 75708873566 101720600746 131962627319 44909629158

Листинг:

```
from decimal import Decimal

N1 = 359690807803
N2 = 361062169537
N3 = 363514381513

C1 = '''
177412278620
8631904062
60910035474
297496979396
44306701511
223949114264
95163574676
126740768642
306466049596
82343556476
97754924718
242675823829
'''

C2 = '''
227891126441
175684889961
108275398403
50799922679
50861774819
120598551775
214220319631
193858968963
```

```
243446962166
168236630688
260389624172
86845867002
'''

C3 = '''
230974691188
345734293737
118726556071
220369632983
69236028918
121704957571
269179568504
201685371953
75708873566
101720600746
131962627319
44909629158
'''


def get_module(N1, N2, N3):
    m1 = N2 * N3
    m2 = N1 * N3
    m3 = N1 * N2
    return m1, m2, m3


def get_inverses(m1, m2, m3, N1, N2, N3):
    n1 = pow(m1, -1, N1)
    n2 = pow(m2, -1, N2)
    n3 = pow(m3, -1, N3)
    return n1, n2, n3


def decrypt_block_crt(block1, block2, block3, n1, n2, n3, m1, m2, m3,
M0):
    s = (block1 * n1 * m1)
    s += (block2 * n2 * m2)
    s += (block3 * n3 * m3)
    x = s % M0
    message_block = round(pow(x, (1/3)))
    return message_block.to_bytes(4, byteorder='big').decode('cp1251')


def decrypt_message_crt(N1, N2, N3, ciphertext1, ciphertext2,
ciphertext3):
    blocks1 = list(map(int, ciphertext1.split()))
    blocks2 = list(map(int, ciphertext2.split()))
    blocks3 = list(map(int, ciphertext3.split()))
    decrypted_message = ""

    M0 = N1 * N2 * N3
    n1, n2, n3 = get_module(N1, N2, N3)
```

```
    m1, m2, m3 = get_inverses(n1, n2, n3, N1, N2, N3)

    for i in range(len(blocks1)):
        decrypted_part = decrypt_block_crt(blocks1[i], blocks2[i],
blocks3[i], m1, m2, m3, n1, n2, n3, M0)
        decrypted_message += decrypted_part
    return decrypted_message


decrypted_message = decrypt_message_crt(N1, N2, N3, C1, C2, C3)
print(decrypted_message)
```

Результаты программы:

```
Протоколы определяют синтаксис-семантику данных.
```