Федеральное государственное автономное образовательное учреждение высшего образования

Университет ИТМО

Дисциплина: Информационная безопасность (Криптография)

Лабораторная работа 2.4

Вариант 12

Работу выполнил студент группы P34111:
Кривоносов Егор Дмитриевич

Преподаватель:
Маркина Татьяна Анатольевна

2022 г.

г. Санкт-Петербург

Оглавление

Цель работы

Изучить атаку на алгоритм шифрования RSA посредством Китайской теоремы об остатках.

Задание

Вариант	**Модуль**			**Блоки зашифрованного текста**		
	N_1	N_2	N_3	C_1	C_2	C_3
12	473302960111	476210148031	478258728547	384927940677 473049749478 98141220439 47772742554 85402795076 49762300554 243238759870 132174590679 394107604075 292566652796 394413369679 176379334217 425745574767 279970734890	47337377053 15502694428 81559584886 360290532716 378412185459 471133458035 276394936545 2116712669 37111299200 387986386867 97786707059 256442600412 455327955288 119517607360	342954751710 440889851539 67503329756 462595462377 84092175909 57911552136 60433527302 25311956275 370327609107 296462225245 241699085506 465708091819 454345671530 210180151910

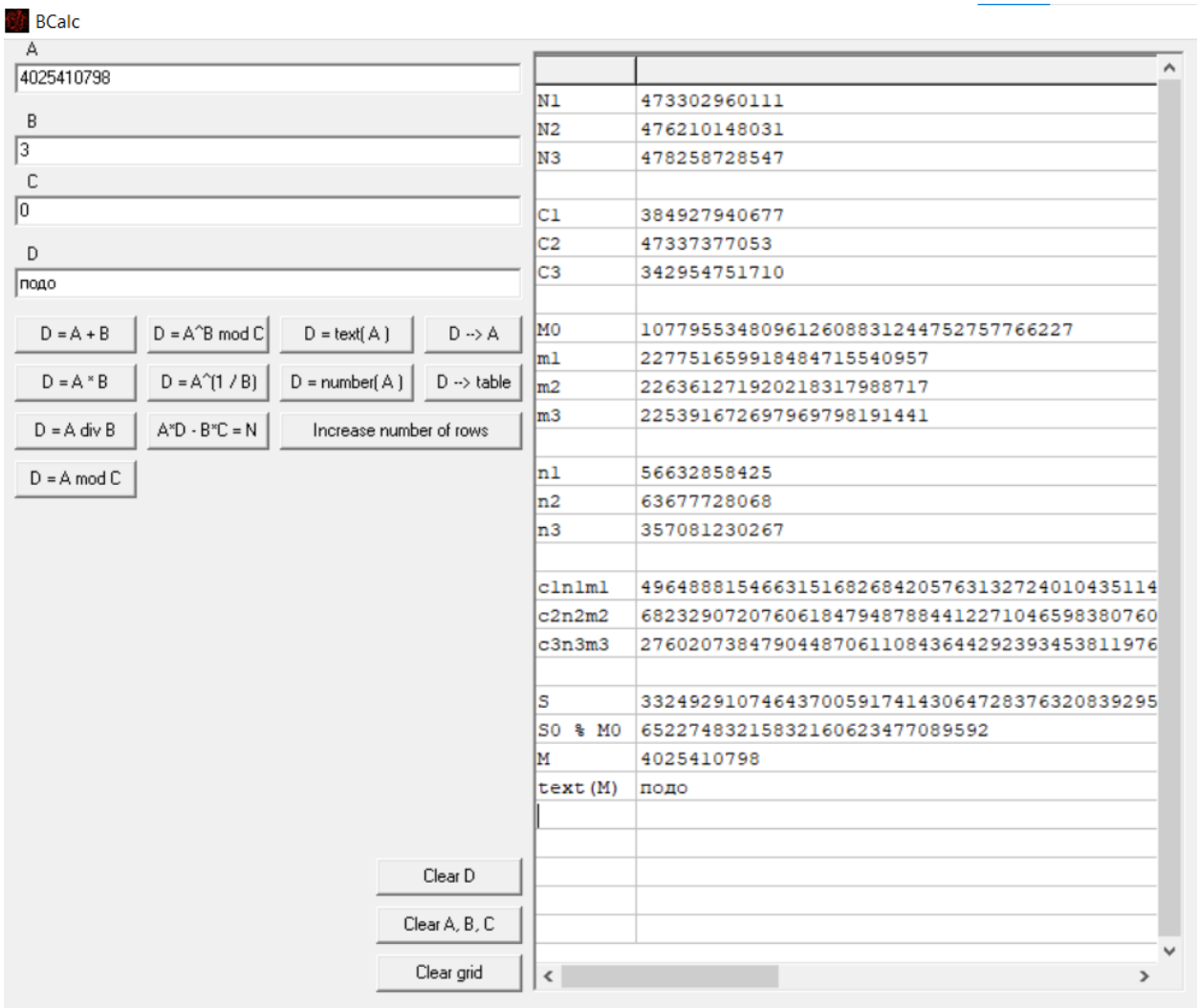
Экспонента для варианта задание $e = 3$

Ход работы

1. Последовательно вычисляем параметры:
   a. $M_0 = N_1 \cdot N_2 \cdot N_3 = 107795534809612608831244752757766227$
   b. $m_1 = N_2 \cdot N_3 = 227751659918484715540957$
   c. $m_2 = N_1 \cdot N_3 = 226361271920218317988717$
   d. $m_3 = N_1 \cdot N_2 = 225391672697969798191441$
   e. $n_1 = m_1^{-1} mod\ N_1 = 56632858425$
   f. $n_2 = m_2^{-1} mod\ N_2 = 63677728068$
   g. $n_3 = m_2^{-1} mod\ N_2 = 357081230267$

2. Вычисляем значение для блока зашифрованного текста
   $S = c_1 \cdot n_1 \cdot m_1 + c_2 \cdot n_2 \cdot m_2 + c_2 \cdot n_2 \cdot m_2$
3. Вычисляем значение $M = (S \bmod M_0)^{1/3}$
4. Преобразуем результат в текст.
5. Повторяем шаги 2-5 для каждой строки.

Скриншот работы программы Bcalc.exe



Листинг разработанной программы

```
from decimal import Decimal

N1 = 473302960111
N2 = 476210148031
N3 = 478258728547

C1 = '''
384927940677
473049749478
98141220439
47772742554
85402795076
49762300554
243238759870
132174590679
394107604075
292566652796
394413369679
176379334217
425745574767
279970734890
'''

C2 = '''
47337377053
15502694428
81559584886
360290532716
378412185459
471133458035
276394936545
2116712669
37111299200
387986386867
97786707059
256442600412
455327955288
119517607360
'''

C3 = '''
342954751710
440889851539
67503329756
462595462377
84092175909
57911552136
60433527302
25311956275
370327609107
296462225245
241699085506
465708091819
454345671530
210180151910
'''
```

```python
def solver(N1, N2, N3, C1, C2, C3):
    print("==============================")
    print("=========== DATA ============")
    print("==============================")
    print(f"N1 = {N1}")
    print(f"N2 = {N2}")
    print(f"N3 = {N3}")
    print(f"C1 = {C1}")
    print(f"C2 = {C2}")
    print(f"C3 = {C3}")

    print("==============================")
    print("== SOLVE - Chinese theorem ==")
    print("==============================")

    c1 = list(map(int, C1.split()))
    c2 = list(map(int, C2.split()))
    c3 = list(map(int, C3.split()))
    message = ""

    M0 = N1 * N2 * N3
    m1 = N2 * N3
    m2 = N1 * N3
    m3 = N1 * N2
    n1 = pow(m1, -1, N1)
    n2 = pow(m2, -1, N2)
    n3 = pow(m3, -1, N3)

    print(f"M0 = N1 * N2 * N3 = {N1} * {N2} * {N3} = {M0}", "\n")
    print(f"m1 = N2 * N3 = {N2} * {N3} = {m1}")
    print(f"m2 = N1 * N3 = {N1} * {N3} = {m2}")
    print(f"m3 = N1 * N2 = {N1} * {N2} = {m3}", "\n")
    print(f"n1 = m1^(-1) mod N1 = {m1}^(-1) mod {N1} = {n1}")
    print(f"n2 = m2^(-1) mod N2 = {m2}^(-1) mod {N2} = {n2}")
    print(f"n3 = m3^(-1) mod N3 = {m3}^(-1) mod {N3} = {n3}", "\n")

    for i in range(len(c1)):
        S = (c1[i] * n1 * m1) + (c2[i] * n2 * m2) + (c3[i] * n3 * m3)
        SmodM0 = S % M0
        M = round(SmodM0 ** (Decimal(1 / 3)))
        part = M.to_bytes(4, byteorder='big').decode('cp1251')
        message += part
        print(f"S[{i}] = c1[{i}]*n1*m1 + c2[{i}]*n2*m2 + c3[{i}]*n3*m3 = {S}")
        print(f"SmodM0[{i}] = S[{i}] mod M0 = {SmodM0}")
        print(f"M = SmodM0[{i}]^(1/3) = {M} => text(M) = {part}", "\n")

    print(f"message = {message}")


solver(N1, N2, N3, C1, C2, C3)
```

Скриншоты работы программы Python

```
============================
========== DATA ============
============================
N1 = 473302960111
N2 = 476210148031
N3 = 478258728547
C1 =
384927940677
473049749478
98141220439
47772742554
85402795076
49762300554
243238759870
132174590679
394107604075
292566652796
394413369679
176379334217
425745574767
279970734890
```

```
C2 =
47337377053
15502694428
81559584886
360290532716
378412185459
471133458035
276394936545
2116712669
37111299200
387986386867
97786707059
256442600412
455327955288
119517607360
```

```
C3 =
342954751710
440889851539
67503329756
462595462377
84092175909
57911552136
60433527302
25311956275
370327609107
296462225245
241699085506
465708091819
454345671530
210180151910

============================
== SOLVE - Chinese theorem ==
============================
M0 = N1 * N2 * N3 = 473302960111 * 476210148031 * 478258728547 = 107795534809612608831244752757766227

m1 = N2 * N3 = 476210148031 * 478258728547 = 227751659918484715540957
m2 = N1 * N3 = 473302960111 * 478258728547 = 226361271920218317988717
m3 = N1 * N2 = 473302960111 * 476210148031 = 225391672697969798191441
```

```
n1 = m1^(-1) mod N1 = 227751659918484715540957^(-1) mod 473302960111 = 56632858425
n2 = m2^(-1) mod N2 = 226361271920218317988717^(-1) mod 476210148031 = 63677728068
n3 = m3^(-1) mod N3 = 225391672697969798191441^(-1) mod 478258728547 = 357081230267

S[0] = c1[0]*n1*m1 + c2[0]*n2*m2 + c3[0]*n3*m3 = 33249291074643700591741430647283763208392958263
SmodM0[0] = S[0] mod M0 = 652274832158321606234770895 92
M = SmodM0[0]^(1/3) = 4025410798 => text(M) = подо

S[1] = c1[1]*n1*m1 + c2[1]*n2*m2 + c3[1]*n3*m3 = 418091595748152710335499341780331813264514347 51
SmodM0[1] = S[1] mod M0 = 544599031291289494134177553 91
M = SmodM0[1]^(1/3) = 3790463231 => text(M) = 6ная

S[2] = c1[2]*n1*m1 + c2[2]*n2*m2 + c3[2]*n3*m3 = 787434128935267135887914588905058025254913982 3
SmodM0[2] = S[2] mod M0 = 168859955225098304125215048
M = SmodM0[2]^(1/3) = 552724722 => text(M) =  сит

S[3] = c1[3]*n1*m1 + c2[3]*n2*m2 + c3[3]*n3*m3 = 430406066465949736197942479207730272840211445 65
SmodM0[3] = S[3] mod M0 = 68498594807201486009395776000
M = SmodM0[3]^(1/3) = 4091606760 => text(M) = уаци
```

```
S[4] = c1[4]*n1*m1 + c2[4]*n2*m2 + c3[4]*n3*m3 = 133240448385816737908287562983113198650079211 27
SmodM0[4] = S[4] mod M0 = 784219413932487066393363042 64
M = SmodM0[4]^(1/3) = 4280349154 => text(M) = я св

S[5] = c1[5]*n1*m1 + c2[5]*n2*m2 + c3[5]*n3*m3 = 120937472600099705164930989089161311256325533702
SmodM0[5] = S[5] mod M0 = 5965341841840055055496565485 6
M = SmodM0[5]^(1/3) = 3907315186 => text(M) = идет

S[6] = c1[6]*n1*m1 + c2[6]*n2*m2 + c3[6]*n3*m3 = 11985232670234611318067127486350385468437916364
SmodM0[6] = S[6] mod M0 = 57398467964121855547794721233
M = SmodM0[6]^(1/3) = 3857448177 => text(M) = ельс

S[7] = c1[7]*n1*m1 + c2[7]*n2*m2 + c3[7]*n3*m3 = 377251421508998083708473119903873287112825746 4
SmodM0[7] = S[7] mod M0 = 67665921498096193958738276125
M = SmodM0[7]^(1/3) = 4074959845 => text(M) = твуе

S[8] = c1[8]*n1*m1 + c2[8]*n2*m2 + c3[8]*n3*m3 = 3542334542004241984921324353150661169054886050 4
SmodM0[8] = S[8] mod M0 = 671162508988538915234823986 71
M = SmodM0[8]^(1/3) = 4063895791 => text(M) = т: п

S[9] = c1[9]*n1*m1 + c2[9]*n2*m2 + c3[9]*n3*m3 = 33226303105098034071923364845394887861480174567
SmodM0[9] = S[9] mod M0 = 64427957598391047551664123904
M = SmodM0[9]^(1/3) = 4008895984 => text(M) = отер
```

```
S[10] = c1[10]*n1*m1 + c2[10]*n2*m2 + c3[10]*n3*m3 = 259494480601781689868543721167418334501861108 61
SmodM0[10] = S[10] mod M0 = 784219131416704460035645440 00
M = SmodM0[10]^(1/3) = 4280348640 => text(M) = я па

S[11] = c1[11]*n1*m1 + c2[11]*n2*m2 + c3[11]*n3*m3 = 434530359953782790030260006241667396953306435 90
SmodM0[11] = S[11] mod M0 = 61206700082427061652239191352
M = SmodM0[11]^(1/3) = 3940938478 => text(M) = кето

S[12] = c1[12]*n1*m1 + c2[12]*n2*m2 + c3[12]*n3*m3 = 48621702902655796825326238345957797834445964713
SmodM0[12] = S[12] mod M0 = 54781128070979442116678545951
M = SmodM0[12]^(1/3) = 3797901151 => text(M) = в___

S[13] = c1[13]*n1*m1 + c2[13]*n2*m2 + c3[13]*n3*m3 = 22249831231546813521103020985899092885026036180
SmodM0[13] = S[13] mod M0 = 40649051007622396560204431 36
M = SmodM0[13]^(1/3) = 1595940896 => text(M) = _

message = подобная ситуация свидетельствует: потеря пакетов____
```

Полученный результат: “подобная ситуация свидетельствует: потеря пакетов____ ”

Вывод

В ходе выполнения данной лабораторной работы я ознакомился с методом для атаки на алгоритм шифрования RSA, основанном на Китайской теореме об остатках.

Полезные ссылки

[Библиотека Decimal]

$x = y^e mod\ z$