Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Информационная безопасность

Лабораторная работа №3
Атака на алгоритм шифрования RSA методом бесключевого чтения

Вариант 14

Студент: Мокров Семён Андреевич

P34121

Преподаватель: Маркина Татьяна Анатольевна

Санкт-Петербург 2023

Содержание

Цель работы

Изучить атаку на алгоритм шифрования RSA посредством метода бесключевого чтения.

Задание

– ознакомьтесь с теорией в [3], в подразделе («Бесключевое чтение»);
– получите вариант задания у преподавателя;
– по полученным данным определите значения r и s при условии, чтобы e1*r – e2*s =1. Для этого необходимо использовать расширенный алгоритм Евклида;
– используя полученные выше значения r и s, запишите исходный текст;
– результаты и промежуточные вычисления значений для любых трех блоков шифрованного текста оформите в виде отчета.

Вариант задания

14	573308195401	973169	550351	327707922480	484439401392
				455697659443	92203619034
				469317095774	199299165882
				41173012855	100840467257
				95114431187	42877265767
				183548202066	537319004931
				114278917224	212469277565
				111319924653	335238563578
				302320646938	215934710265
				497834611165	248375790884
				207393954597	8143413999
				469317095774	199299165882
				184588110993	484325656679

Листинг разработанной программы

```
from numpy.core.defchararray import isnumeric
from termcolor import colored

N = 573308195401

e_1 = 973169
e_2 = 550351

C_1 = '''
327707922480
455697659443
469317095774
41173012855
95114431187
183548202066
114278917224
111319924653
302320646938
497834611165
207393954597
469317095774
184588110993
'''

C_2 = '''
484439401392
92203619034
199299165882
100840467257
42877265767
537319004931
212469277565
335238563578
215934710265
248375790884
8143413999
```

```
199299165882
484325656679
'''

print(colored("Стартовые данные:", "green"))
print("C1: " + C_1)
print("C2: " + C_2)
print("N: " + str(N))
print("e1: " + str(e_1))
print("e2: " + str(e_2))


def get_int_list(source_list):
    result = []
    for curr_str in source_list.split():
        if isnumeric(curr_str):
            result.append(int(curr_str))
    return result


# Расширенный алгоритм Евклида
def extended_gcd(a, b):
    if a == 0:
        return 0, 1
    else:
        x_arg, y_arg = extended_gcd(b % a, a)
    return y_arg - (b // a) * x_arg, x_arg


C_1 = get_int_list(C_1)
C_2 = get_int_list(C_2)
result = ""

x, y = extended_gcd(e_1, e_2)

print(colored("Параметры, полученные расширенным алгоритмом
Евклида", "green"))
```

```
print("r = " + str(x))
print("s = " + str(y))

print(colored("\nХод вычислений: ", "green"))

for i in range(len(C_1)):
    C_1_x = pow(C_1[i], x, N)
    C_2_y = pow(C_2[i], y, N)
    m = (C_1_x * C_2_y) % N
    current_result_bytes = int.to_bytes(m, length=4, byteorder='big')
    current_part = current_result_bytes.decode('windows-1251')
    result += current_part

    print("C1[", str(i), "]^r) mod N =", str(C_1_x))
    print("C2[", str(i), "]^s) mod N =", str(C_2_y))
    print("Полученная часть сообщения: ", str(current_part), "\n")

print(colored("RESULT = ", "green") + result)
```

Исходный код

Исходный код расположен в репозитории:
https://github.com/semwett0301/information-security

Результаты работы программы

Скриншоты вывода программы

```
Стартовые данные:
C1:
327707922480
455697659443
469317095774
41173012855
95114431187
183548202066
114278917224
111319924653
302320646938
497834611165
207393954597
469317095774
184588110993

C2:
484439401392
92203619034
199299165882
100840467257
42877265767
537319004931
212469277565
335238563578
215934710265
248375790884
8143413999
199299165882
484325656679

N: 573308195401
e1: 973169
e2: 550351
```

```
Параметры, полученные расширенным алгоритмом Евклида
r = 152315
s = -269334
```

```
Ход вычислений:
C1[ 0 ]^r) mod N = 165024469333
C2[ 0 ]^s) mod N = 423909999356
Полученная часть сообщения:  льки

C1[ 1 ]^r) mod N = 299944833361
C2[ 1 ]^s) mod N = 509228750708
Полученная часть сообщения:  ми п

C1[ 2 ]^r) mod N = 444024439693
C2[ 2 ]^s) mod N = 295389020807
Полученная часть сообщения:  акет

C1[ 3 ]^r) mod N = 168070732157
C2[ 3 ]^s) mod N = 206311219044
Полученная часть сообщения:  ами.

C1[ 4 ]^r) mod N = 277377816945
C2[ 4 ]^s) mod N = 492544202182
Полученная часть сообщения:   Пер

C1[ 5 ]^r) mod N = 351512795454
C2[ 5 ]^s) mod N = 357761672628
Полученная часть сообщения:  вым

C1[ 6 ]^r) mod N = 551962777586
C2[ 6 ]^s) mod N = 111237883233
Полученная часть сообщения:  сред

C1[ 7 ]^r) mod N = 57399190242
C2[ 7 ]^s) mod N = 151437442937
Полученная часть сообщения:  и ни

C1[ 8 ]^r) mod N = 35384552387
C2[ 8 ]^s) mod N = 405541098220
Полученная часть сообщения:  х яв
```

```
C1[ 9 ]^r) mod N = 550170293060
C2[ 9 ]^s) mod N = 2208188518
Полученная часть сообщения:  ляет

C1[ 10 ]^r) mod N = 96973721996
C2[ 10 ]^s) mod N = 246877745429
Полученная часть сообщения:  ся п

C1[ 11 ]^r) mod N = 444024439693
C2[ 11 ]^s) mod N = 295389020807
Полученная часть сообщения:  акет

C1[ 12 ]^r) mod N = 275388717443
C2[ 12 ]^s) mod N = 174889350036
Полученная часть сообщения:
```

```
RESULT = лькими пакетами. Первым среди них является пакет
```

Полученное сообщение

лькими пакетами. Первым среди них является пакет

Выводы

В данной лабораторной работе я:

- Ознакомился с принципом взлома RSA при помощи метода бесключевого чтения
- Реализовал процесс взлома на Python