МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

Кафедра ИИТ

ОТЧЁТ
По лабораторной работе №3
«Атака на алгоритм шифрования RSA посредством метода бесключевого чтения»

Выполнил:
Студент группы ИИ-22
Дубина Н.С.
Проверила:
Хацкевич А.С.

Брест 2024

Цель работы: изучить атаку на алгоритм шифрования RSA посредством метода бесключевого чтения

Ход работы

Вариант	Модуль, N	Экспонента		Блок зашифрованного текста	
		e_1	e_2	C_1	C_2
20	502110569407	693661	366287	451590415251 110439571420 183752091528 274872936616 28541011195 450835617776 260759622383 342128341762 158761845107 190701543235 336633436793 107036107438 143086295492	489035727840 352254618578 112984103119 324252397833 258279989467 309371933868 309370695834 275718202556 484547254614 319090281932 321505940571 499673648361 445389404030

Код программы:

```
# Заданные параметры
N = 502110569407
e1 = 693661
e2 = 366287

# Списки зашифрованных значений
C1 = [
    451590415251,
    110439571420,
    183752091528,
    274872936616,
    28541011195,
    450835617776,
    260759622383,
    342128341762,
    158761845107,
    190701543235,
    336633436793,
    107036107438,
    143086295492
]

C2 = [
    489035727840,
    352254618578,
    112984103119,
    324252397833,
    258279989467,
    309371933868,
    309370695834,
    275718202556,
    484547254614,
    319090281932,
    321505940571,
    499673648361,
    445389404030
]


def extended_gcd(a, b):
    """Расширенный алгоритм Евклида для нахождения коэффициентов r и s"""
    old_r, r = a, b
    old_s, s = 1, 0
    old_t, t = 0, 1
```

```
    while r != 0:
        quotient = old_r // r
        old_r, r = r, old_r - quotient * r
        old_s, s = s, old_s - quotient * s
        old_t, t = t, old_t - quotient * t

    # old_r — это gcd, old_s и old_t — коэффициенты, такие что a*old_s + b*old_t = gcd
    return old_r, old_s, old_t


def main():
    # Проверяем gcd и находим коэффициенты r и s
    gcd_value, r, s = extended_gcd(e1, e2)

    if gcd_value != 1:
        raise ValueError("Не удалось найти коэффициенты r и s, так как gcd(e1, e2) != 1.")

    print(f"Найденные коэффициенты: r = {r}, s = {s}")

    # Перебираем все блоки зашифрованного текста
    for index in range(len(C1)):
        print(f"\nДешифрование блока {index + 1}:")
        # Дешифрование
        # Вычисляем C1^r mod N
        c1_r = pow(C1[index], r, N)

        # Вычисляем C2^(-s) mod N
        c2_s_inv = pow(pow(C2[index], -1, N), s, N)

        # Перемножаем результаты и берём модуль N
        m = (c1_r * c2_s_inv) % N

        # Преобразуем результат в текст
        print("Дешифрованное сообщение m:", m)
        try:
            print("Исходный текст:", m.to_bytes((m.bit_length() + 7) // 8, 'big').decode('windows-1251'))
        except UnicodeDecodeError:
            print("Не удалось декодировать текст. Возможно, сообщение в другой кодировке.")


if __name__ == "__main__":
    main()
```

Результат работы:

```
Найденные коэффициенты: r = -69223, s = 131092

Дешифрование блока 1:
Дешифрованное сообщение m: 357715725983
Исходный текст: SI„Oų
```

Вывод: изучил атаку на алгоритм шифрования RSA посредством метода бесключевого чтения