Тема 7. Асимметричные шифры. Аналог шифра RSA

Теоретическая часть

Приведена в сопровождающей лекции

Практическая часть

Составить компьютерную программу (на любом языке программирования), которая реализует уменьшённый аналог шифра RSA. Программа должна выполнять следующие действия (этапы работы).

3. **Создание ключей.** Секретный ключ SK и публичный ключ PK представляют собой пару целых чисел.
4. **Шифрование сообщения.** Сообщение, являющееся целым числом, преобразуется в другое целое число.
5. **Расшифрование сообщения.** Сообщение, являющееся целым числом, преобразуется в другое целое число.

В строчках ниже можно видеть пример того, как реализуется указанный подход

```
# подготовительная часть
import math
import random

# вычисление обратного к числу a по модулю n
def inv(a, n):
    if a % n == 0:
        return None
    else:
        a_inv = 1
        while True:
            if (a_inv * a) % n == 1:
                return a_inv
            a_inv += 1

# вычисление функции Эйлера
def phi(n):
    s = 0
    for i in range(n):
        if math.gcd(n, i) == 1:
            s += 1
    return s

# вычисление ключей
def gen_pk_sk(p, q):
    n = p * q
    ph = phi(n)
    # значение параметра e может быть получено случайно
    ex = [_ for _ in range(3, ph) if math.gcd(_, ph) == 1]
    e = ex[random.randint(0, len(ex) - 1)]
    # либо значение e может быть задано явно
    # e = 23
    d = inv(e, ph)
    pk = (e, n)
    sk = (d, n)
    return pk, sk

# шифрование
def chifering(m, keys):
    pk = keys[0]
    # sk = keys[1]
```

```
    e = pk[0]
    n = pk[1]
    # d = sk[0]
    c = (m ** e) % n
    return c

# расшифрование
def dechifering(c, keys):
    pk = keys[0]
    sk = keys[1]
    # e = pk[0]
    n = pk[1]
    d = sk[0]
    m = (c ** d) % n
    return m

# основная часть
keys = gen_pk_sk(13, 17)
print(f"Ключи: {keys}")
ms = 113
print(f"Исходное сообщение: {ms}")
ch = chifering(ms, keys)
print(f"Зашифрованное сообщение: {ch}")
xms = dechifering(ch, keys)
print(f"Расшифрованное сообщение: {xms}")
```

Окно работающей программы:

```
Ключи: ((25, 221), (169, 221))
Исходное сообщение: 113
Зашифрованное сообщение: 74
Расшифрованное сообщение: 113
```