Криптосистема Хилла. Расшифровывание.

Расшифровка выполняется по формуле P = A^{-1} * C (mod m). A^{-1} – обратная к A матрица по mod m.

$C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$ – вектор; символы, которые хотим расшифровать.

Для матрицы $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ определитель D = det(A) = a*d – b*c. Если gcd (D, m) == 1, то можно посчитать обратную матрицу:

$A^{-1} = D^{-1} * \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$, где D^{-1} – обратное значение по умножению к D по модулю m.

То есть D^{-1} считается по расширернному алгоритму Евклида (D^{-1} = findModInverse(abs(D), m)):

```python
def findModInverse(a, m):
    # Returns the modular inverse of a % m, which is
    # the number x such that a*x % m = 1

    if gcd(a, m) != 1:
        return None

    # Calculate using the Extended Euclidean Algorithm:
    u1, u2, u3 = 1, 0, a
    v1, v2, v3 = 0, 1, m
    print(' ', v1, v2, v3, u1, u2, u3)
    while v3 != 0:
        q = u3 // v3  # // is the integer division operator
        v1, v2, v3, u1, u2, u3 = (u1 - q * v1), (u2 - q * v2), (u3 - q * v3), v1, v2, v3
        print(q, v1, v2, v3, u1, u2, u3)
    return u1 % m
```

Так как A^{-1} – обратная к A матрица по mod m, то $A^{-1} = \begin{bmatrix} (d * D^{-1}) \bmod m & (-b * D^{-1}) \bmod m \\ (-c * D^{-1}) \bmod m & (a * D^{-1}) \bmod m \end{bmatrix}$.

Соответственно в формуле P = A^{-1} * C (mod m), по которой выполняется расшифровка, тоже все элементы должны быть по mod m. P = $\begin{bmatrix} (d * D^{-1}) \bmod m & (-b * D^{-1}) \bmod m \\ (-c * D^{-1}) \bmod m & (a * D^{-1}) \bmod m \end{bmatrix} * \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} P_1 \bmod m \\ P_2 \bmod m \end{bmatrix}$.

Пример, который в файле Занятие 2.pdf:

Пусть есть $A = \begin{bmatrix} 5 & 17 \\ 4 & 15 \end{bmatrix}$, m = 26.

Вычисляем D = 5*15 – 17*4 = 7. – Если здесь получается отрицательное число, то берем его абсолютную величину: D = abs(D).

Вычисляем D^{-1} = <по расширенному алгоритму Евклида> = 15.

$A^{-1} = \begin{bmatrix} (15 * 15) \bmod 26 & (-17 * 15) \bmod 26 \\ (-4 * 15) \bmod 26 & (5 * 15) \bmod 26 \end{bmatrix} = \begin{bmatrix} 225 \bmod 26 & -255 \bmod 26 \\ -60 \bmod 26 & 75 \bmod 26 \end{bmatrix} = \begin{bmatrix} 17 & 5 \\ 18 & 23 \end{bmatrix}$.

Задание 0.

Расшифровать картинку, если известен ключ K = $\begin{bmatrix} 189 & 58 \\ 21 & 151 \end{bmatrix}$.

Нужно найти обратную матрицу для K по mod 256.

$K^{-1} = \begin{bmatrix} 207 & 246 \\ 195 & 37 \end{bmatrix}$.

Далее перебираем **пары значений** картинки с шагом 2. И для пар значений вычисляем –

$K^{-1} * \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} (207 * C_1 + 246 * C_2)\ mod\ 256 \\ (195 * C_1 + 37 * C_2)\ mod\ 256 \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$, где P_1 и P_2 – расшифрованные значения.

Задание 1.

Есть зашифрованная картинка. Известно, что первые четыре байта в **расшифрованной** картинке имеют значения – 137, 80, 78, 71.

Посмотрим первые четыре байта у **зашифрованной** картинки: 23, 3, 239, 52.

Чтобы расшифровать полностью картинку, нужен ключ. Найдем его. Пусть K = $\begin{bmatrix} K_1 & K_2 \\ K_3 & K_4 \end{bmatrix}$ – искомый ключ.

Тогда, картинка шифровалась следующим образом:

$\begin{bmatrix} K_1 & K_2 \\ K_3 & K_4 \end{bmatrix} * \begin{bmatrix} 137 \\ 80 \end{bmatrix} = \begin{bmatrix} 23 \\ 3 \end{bmatrix}$ и $\begin{bmatrix} K_1 & K_2 \\ K_3 & K_4 \end{bmatrix} * \begin{bmatrix} 78 \\ 71 \end{bmatrix} = \begin{bmatrix} 239 \\ 52 \end{bmatrix}$.

Распишем в линейные уравнения:

$\begin{cases} K_1 * 137 + K_2 * 80 = 23 \\ K_3 * 137 + K_4 * 80 = 3 \end{cases}$ и $\begin{cases} K_1 * 78 + K_2 * 71 = 239 \\ K_3 * 78 + K_4 * 71 = 52 \end{cases}$.

Теперь сгруппируем:

$\begin{cases} K_1 * 137 + K_2 * 80 = 23 \\ K_1 * 78 + K_2 * 71 = 239 \end{cases}$ и $\begin{cases} K_3 * 137 + K_4 * 80 = 3 \\ K_3 * 78 + K_4 * 71 = 52 \end{cases}$

То есть получили:

$\begin{bmatrix} 137 & 80 \\ 78 & 71 \end{bmatrix} * \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} 23 \\ 239 \end{bmatrix}$ и $\begin{bmatrix} 137 & 80 \\ 78 & 71 \end{bmatrix} * \begin{bmatrix} K_3 \\ K_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 52 \end{bmatrix}$

Выражаем $\begin{bmatrix} K_i \\ K_j \end{bmatrix}$. Домножаем обе части равенства на матрицу обратную к $\begin{bmatrix} 137 & 80 \\ 78 & 71 \end{bmatrix}$ по mod 256.

Обратная матрица = $\begin{bmatrix} 89 & 80 \\ 175 & 251 \end{bmatrix}$

$\begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} 89 & 80 \\ 175 & 251 \end{bmatrix} \begin{bmatrix} 23 \\ 239 \end{bmatrix} = \begin{bmatrix} 175 \\ 251 \end{bmatrix}$ и $\begin{bmatrix} K_3 \\ K_4 \end{bmatrix} = \begin{bmatrix} 89 & 80 \\ 175 & 251 \end{bmatrix} \begin{bmatrix} 3 \\ 52 \end{bmatrix} = \begin{bmatrix} 75 \\ 214 \end{bmatrix}$.

В итоге получили матрицу: $\begin{bmatrix} 175 & 251 \\ 75 & 214 \end{bmatrix}$ – это и есть наш искомый ключ.

Проверим. Возьмем первые два значения картинки: 137, 80.

$\begin{bmatrix} 175 & 251 \\ 75 & 214 \end{bmatrix} * \begin{bmatrix} 137 \\ 80 \end{bmatrix} = \begin{bmatrix} 23 \\ 3 \end{bmatrix}$ (mod 256).

Задание 2.

Аналогично заданию 1. Только известно, что есть файл, который начинается на слово Whose.

Достаточно получить значения первых четырех букв W, h, o, s из таблицы символов ASCII.

В Python это можно сделать с помощью функции ord(). (ord(‘W’)).

И повторить все вычисления из задания 1.