Федеральное государственное автономное образовательное учреждение высшего образования “Национальный исследовательский университет ИТМО”

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №2

“ Основы шифрования данных”

из раздела

“Криптографические системы с секретным ключом”

по дисциплине

“Информационная безопасность”

Вариант 11

Выполнил:

Иванов Матвей Сергеевич

группа P34111

Преподаватель:

Маркина Татьяна Анатольевна

г. Санкт-Петербург, 2024

Цель работы:

Изучение структуры и основных принципов работы современных алгоритмов блочного симметричного шифрования, приобретение навыков программной реализации блочных симметричных шифров.

Листинг программы:

```
import os
from string import printable
import numpy as np

# Consts
rus =
"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯабвгдеёжзийклмнопрстуфхцчшщъ
ыьэюя«»—"
alph = rus + printable


Sbox = np.array([
    0x63, 0x7C, 0x77, 0x7B, 0xF2, 0x6B, 0x6F, 0xC5, 0x30,
0x01, 0x67, 0x2B, 0xFE, 0xD7, 0xAB, 0x76,
    0xCA, 0x82, 0xC9, 0x7D, 0xFA, 0x59, 0x47, 0xF0, 0xAD,
0xD4, 0xA2, 0xAF, 0x9C, 0xA4, 0x72, 0xC0,
    0xB7, 0xFD, 0x93, 0x26, 0x36, 0x3F, 0xF7, 0xCC, 0x34,
0xA5, 0xE5, 0xF1, 0x71, 0xD8, 0x31, 0x15,
    0x04, 0xC7, 0x23, 0xC3, 0x18, 0x96, 0x05, 0x9A, 0x07,
0x12, 0x80, 0xE2, 0xEB, 0x27, 0xB2, 0x75,
    0x09, 0x83, 0x2C, 0x1A, 0x1B, 0x6E, 0x5A, 0xA0, 0x52,
0x3B, 0xD6, 0xB3, 0x29, 0xE3, 0x2F, 0x84,
    0x53, 0xD1, 0x00, 0xED, 0x20, 0xFC, 0xB1, 0x5B, 0x6A,
0xCB, 0xBE, 0x39, 0x4A, 0x4C, 0x58, 0xCF,
    0xD0, 0xEF, 0xAA, 0xFB, 0x43, 0x4D, 0x33, 0x85, 0x45,
0xF9, 0x02, 0x7F, 0x50, 0x3C, 0x9F, 0xA8,
    0x51, 0xA3, 0x40, 0x8F, 0x92, 0x9D, 0x38, 0xF5, 0xBC,
0xB6, 0xDA, 0x21, 0x10, 0xFF, 0xF3, 0xD2,
    0xCD, 0x0C, 0x13, 0xEC, 0x5F, 0x97, 0x44, 0x17, 0xC4,
0xA7, 0x7E, 0x3D, 0x64, 0x5D, 0x19, 0x73,
```

```
    0x60, 0x81, 0x4F, 0xDC, 0x22, 0x2A, 0x90, 0x88, 0x46,
0xEE, 0xB8, 0x14, 0xDE, 0x5E, 0x0B, 0xDB,
    0xE0, 0x32, 0x3A, 0x0A, 0x49, 0x06, 0x24, 0x5C, 0xC2,
0xD3, 0xAC, 0x62, 0x91, 0x95, 0xE4, 0x79,
    0xE7, 0xC8, 0x37, 0x6D, 0x8D, 0xD5, 0x4E, 0xA9, 0x6C,
0x56, 0xF4, 0xEA, 0x65, 0x7A, 0xAE, 0x08,
    0xBA, 0x78, 0x25, 0x2E, 0x1C, 0xA6, 0xB4, 0xC6, 0xE8,
0xDD, 0x74, 0x1F, 0x4B, 0xBD, 0x8B, 0x8A,
    0x70, 0x3E, 0xB5, 0x66, 0x48, 0x03, 0xF6, 0x0E, 0x61,
0x35, 0x57, 0xB9, 0x86, 0xC1, 0x1D, 0x9E,
    0xE1, 0xF8, 0x98, 0x11, 0x69, 0xD9, 0x8E, 0x94, 0x9B,
0x1E, 0x87, 0xE9, 0xCE, 0x55, 0x28, 0xDF,
    0x8C, 0xA1, 0x89, 0x0D, 0xBF, 0xE6, 0x42, 0x68, 0x41,
0x99, 0x2D, 0x0F, 0xB0, 0x54, 0xBB, 0x16
])

SboxInv = np.array([
    0x52, 0x09, 0x6a, 0xd5, 0x30, 0x36, 0xa5, 0x38, 0xbf,
0x40, 0xa3, 0x9e, 0x81, 0xf3, 0xd7, 0xfb,
    0x7c, 0xe3, 0x39, 0x82, 0x9b, 0x2f, 0xff, 0x87, 0x34,
0x8e, 0x43, 0x44, 0xc4, 0xde, 0xe9, 0xcb,
    0x54, 0x7b, 0x94, 0x32, 0xa6, 0xc2, 0x23, 0x3d, 0xee,
0x4c, 0x95, 0x0b, 0x42, 0xfa, 0xc3, 0x4e,
    0x08, 0x2e, 0xa1, 0x66, 0x28, 0xd9, 0x24, 0xb2, 0x76,
0x5b, 0xa2, 0x49, 0x6d, 0x8b, 0xd1, 0x25,
    0x72, 0xf8, 0xf6, 0x64, 0x86, 0x68, 0x98, 0x16, 0xd4,
0xa4, 0x5c, 0xcc, 0x5d, 0x65, 0xb6, 0x92,
    0x6c, 0x70, 0x48, 0x50, 0xfd, 0xed, 0xb9, 0xda, 0x5e,
0x15, 0x46, 0x57, 0xa7, 0x8d, 0x9d, 0x84,
    0x90, 0xd8, 0xab, 0x00, 0x8c, 0xbc, 0xd3, 0x0a, 0xf7,
0xe4, 0x58, 0x05, 0xb8, 0xb3, 0x45, 0x06,
    0xd0, 0x2c, 0x1e, 0x8f, 0xca, 0x3f, 0x0f, 0x02, 0xc1,
0xaf, 0xbd, 0x03, 0x01, 0x13, 0x8a, 0x6b,
    0x3a, 0x91, 0x11, 0x41, 0x4f, 0x67, 0xdc, 0xea, 0x97,
0xf2, 0xcf, 0xce, 0xf0, 0xb4, 0xe6, 0x73,
    0x96, 0xac, 0x74, 0x22, 0xe7, 0xad, 0x35, 0x85, 0xe2,
0xf9, 0x37, 0xe8, 0x1c, 0x75, 0xdf, 0x6e,
```

```
    0x47, 0xf1, 0x1a, 0x71, 0x1d, 0x29, 0xc5, 0x89, 0x6f,
0xb7, 0x62, 0x0e, 0xaa, 0x18, 0xbe, 0x1b,
    0xfc, 0x56, 0x3e, 0x4b, 0xc6, 0xd2, 0x79, 0x20, 0x9a,
0xdb, 0xc0, 0xfe, 0x78, 0xcd, 0x5a, 0xf4,
    0x1f, 0xdd, 0xa8, 0x33, 0x88, 0x07, 0xc7, 0x31, 0xb1,
0x12, 0x10, 0x59, 0x27, 0x80, 0xec, 0x5f,
    0x60, 0x51, 0x7f, 0xa9, 0x19, 0xb5, 0x4a, 0x0d, 0x2d,
0xe5, 0x7a, 0x9f, 0x93, 0xc9, 0x9c, 0xef,
    0xa0, 0xe0, 0x3b, 0x4d, 0xae, 0x2a, 0xf5, 0xb0, 0xc8,
0xeb, 0xbb, 0x3c, 0x83, 0x53, 0x99, 0x61,
    0x17, 0x2b, 0x04, 0x7e, 0xba, 0x77, 0xd6, 0x26, 0xe1,
0x69, 0x14, 0x63, 0x55, 0x21, 0x0c, 0x7d
])

mixcol_matrix = np.array([
    [2, 3, 1, 1],
    [1, 2, 3, 1],
    [1, 1, 2, 3],
    [3, 1, 1, 2]
])

mixcol_matrix_inv = np.array([
    [14, 11, 13, 9],
    [9, 14, 11, 13],
    [13, 9, 14, 11],
    [11, 13, 9, 14]
])

key_lenght_bytes = [16, 24, 32]

# Map max(key len, block len) to number of rounds
rounds_map = {
    4: 10,
    6: 12,
    8: 14
}
```

```
shifts_by_block_len = {
    4: [0, 1, 2, 3],
    6: [0, 1, 2, 3],
    8: [0, 1, 3, 4]
}

DEBUG_MODE = False

# Process

# Help methods

def assert_block_size(block_size: str):
    if not block_size.isdigit():
        print("Block size should be correct integer")
        exit(1)

    block_size = int(block_size)
    if block_size not in rounds_map and block_size // 32 not
in rounds_map:
        print("Incorrect block size")
        exit(1)
    elif block_size > 8:
        block_size = block_size // 32

    return block_size


def prepare_message(m, block_len: int):
    res = []
    try:
        for let in m:
            if DEBUG_MODE:
                print(let, end='')
            res.append(alph.index(let) + 1)
    except ValueError as e:
        print("Unacceptable letter: " + str(e))
        exit(1)
```

```
    remain = block_len * 4 - (len(res) % (block_len * 4))
    if remain < block_len * 4:
        res += [0] * remain
    return np.array(res).reshape((4, -1), order='F')


def get_message(indexes):
    mes = ""
    for ind in indexes:
        if ind == 0:
            break
        mes += alph[ind - 1]
    return mes


def print_matrix(m):
    print("\n".join(["\t".join([f"{i:x}" for i in line]) for
line in m]))


def gmul(a, b):
    p = 0
    for c in range(8):
        if b & 1:
            p ^= a
        a <<= 1
        if a & 0x100:
            a ^= 0x11b
        b >>= 1
    return p

# Key Scheduling

def getRcon(num_rounds: int):
    rcon = np.zeros((num_rounds, 4), dtype='int64')
    rcon[0, 0] = 1
    for i in range(1, num_rounds):
```

```
        if rcon[i - 1, 0] >= 0x80:
            rcon[i, 0] = (rcon[i - 1, 0] * 2) ^ 0x11b
        else:
            rcon[i, 0] = rcon[i - 1, 0] * 2
    return rcon


def KeySchedule(key, num_rounds: int):
    n = key.shape[1]

    res = np.zeros((n * num_rounds, 4), dtype='int64')
    res[:n, :] = key.T
    rcon = getRcon(num_rounds)

    if DEBUG_MODE:
        print("Rcon matrix:")
        print_matrix(rcon)

    for i in range(n, n * num_rounds):
        if i % n == 0:
            res[i] = res[i - n] ^ SubBytes(np.roll(res[i - 1],
-1)) ^ rcon[i // n - 1]
        elif n > 6 and i % n == 4:
            res[i] = res[i - n] ^ SubBytes(res[i - 1])
        else:
            res[i] = res[i - n] ^ res[i - 1]

    return res.T


# Round steps

def SubSelection(sub_array, block_len, cur_step):
    return sub_array[:, block_len * cur_step: block_len *
(cur_step + 1)]
```

```
def SubBytes(state, sbox=Sbox):
    return sbox[state]

def ShiftRows(state, inv: bool = False):
    n = state.shape[0]
    sign = int(inv) * 2 - 1
    return np.array([np.roll(state[i], sign *
shifts_by_block_len[n][i]) for i in range(n)])

def MixColumns(state, matrix):
    res = np.zeros(state.shape, dtype='int64')
    for col in range(state.shape[1]):
        for cur_col in range(state.shape[0]):
            for row in range(state.shape[0]):
                res[cur_col][col] ^=
gmul(matrix[cur_col][row], state[row][col])
    return res

def process_encode_round(state, cicle_key, final_round: bool =
False):
    state = SubBytes(state, Sbox)
    if DEBUG_MODE:
        print("[ENCODE] After Sub Bytes")
        print_matrix(state)

    state = ShiftRows(state)
    if DEBUG_MODE:
        print("[ENCODE] After Shift Rows")
        print_matrix(state)

    if not final_round:
        state = MixColumns(state, mixcol_matrix)
        if DEBUG_MODE:
            print("[ENCODE] After Mix Columns")
            print_matrix(state)

    state = state ^ cicle_key
```

```
    if DEBUG_MODE:
        print("[ENCODE] Round result")
        print_matrix(state)

    return state


def encode(message: str, key: str, block_len: int):
    num_rounds = rounds_map[block_len]
    message_mat = prepare_message(message, block_len)

    if DEBUG_MODE:
        print("\n\nInput matrix:")
        print_matrix(message_mat)

    key_mat = prepare_message(key, len(key) // 4)
    key_sched = KeySchedule(key_mat, num_rounds)

    result = np.zeros(message_mat.shape, dtype='int64')

    for cur_block in range(message_mat.shape[1] // block_len):
        state = SubSelection(message_mat, block_len,
cur_block)

        for cur_round in range(num_rounds):
            state = process_encode_round(
                state,
                SubSelection(key_sched, block_len, cur_round),
                final_round=cur_round == num_rounds - 1
            )

        result[:, block_len * cur_block:block_len * (cur_block
+ 1)] = state

    return result
```

```
def process_decode_round(state, cicle_key, first_round: bool =
False):
    state = state ^ cicle_key

    if DEBUG_MODE:
        print("[DECODE] After Adding Round Key")
        print_matrix(state)

    if not first_round:
        state = MixColumns(state, mixcol_matrix_inv)
        if DEBUG_MODE:
            print("[DECODE] After Mix Columns")
            print_matrix(state)



    state = ShiftRows(state, inv=True)
    if DEBUG_MODE:
        print("[DECODE] After Shift Rows")
        print_matrix(state)

    state = SubBytes(state, SboxInv)
    if DEBUG_MODE:
        print("[DECODE] After SubBytes, round result")
        print_matrix(state)

    return state


def decode(encoded_message, key: str, block_len: int):
    num_rounds = rounds_map[block_len]

    key_mat = prepare_message(key, len(key) // 4)
    key_sched = KeySchedule(key_mat, num_rounds)

    result = np.zeros(encoded_message.shape, dtype='int64')

    for cur_block in range(encoded_message.shape[1] //
block_len):
```

```
        state = SubSelection(encoded_message, block_len,
cur_block)

        for cur_round in range(num_rounds):
            state = process_decode_round(
                state,
                SubSelection(key_sched, block_len, num_rounds
- cur_round - 1),
                first_round=cur_round == 0
            )

        result[:, block_len * cur_block:block_len * (cur_block
+ 1)] = state


    if DEBUG_MODE:
        print("\n\nResult matrix:")
        print_matrix(result)
    result = result.reshape(-1, order='F')

    return get_message(result)

# Input
block_size = assert_block_size(input("Input block size (bits
or words): "))
key = input("Input key: ")

if len(key) not in key_lenght_bytes:
    print("Incorrect key lenght")
    exit(1)

filename = input("Input filename: ")
if not os.path.exists(filename):
     print("File not exists")
     exit(1)

with open(filename, "r") as f:
     message = f.read()
```

```
encoded_message = encode(message, key, block_size)

print("\n\nEncoded:")
print_matrix(encoded_message)

decoded_message = decode(encoded_message, key, block_size)
print("\n\nDecoded:")
print(decoded_message)
```

Результат работы:

Исходный тестовый текст:

Он, однако ж, не то чтоб уж был совсем в беспамятстве во всё время болезни: это было лихорадочное состояние, с бредом и полусознанием. Многое он потом припомнил. То казалось ему, что около него собирается много народу и хотят его взять и куда-то вынести, очень об нем спорят и ссорятся. То вдруг он один в комнате, все ушли и боятся его, и только изредка чуть-чуть отворяют дверь посмотреть на него, грозят ему, сговариваются об чем-то промеж себя, смеются и дразнят его. Настасью он часто помнил подле себя; различал и еще одного человека, очень будто бы ему знакомого, но кого именно - никак не мог догадаться и тосковал об этом, даже и плакал. Иной раз казалось ему, что он уже с месяц лежит; в другой раз - что всё тот же день идет. Но об том - об том он совершенно забыл; зато ежеминутно помнил, что об чем-то забыл, чего нельзя забывать, - терзался, мучился, припоминая, стонал, впадал в бешенство или в ужасный, невыносимый страх. Тогда он порывался с места, хотел бежать, но всегда кто-нибудь его останавливал силой, и он опять впадал в бессилие и беспамятство. Наконец он совсем пришел в себя.

Произошло это утром, в десять часов. В этот час утра, в ясные дни, солнце всегда длинною полосой проходило по его правой стене и освещало угол подле двери. У постели его стояла Настасья и еще один человек, очень любопытно его разглядывавший и совершенно ему незнакомый. Это был молодой парень в кафтане, с бородкой, и с виду походил на артельщика. Из полуотворенной двери выглядывала хозяйка. Раскольников приподнялся.

Размер блока: 256

Ключ шифрования (256 бит): OaoRlgwkYJCdqhxifZbemzjVQtFNpEcn

Закодированное сообщение (в hexdecimal виде):

af	db	8b	9a	70	cd	20	76	2c	70	c1	59	c0
	82	7e	50	50	b9	ea	96	ff	f9	da	8e	23
	e1	40	69	ff	36	28	a4	a3	a9	3d	d4	c1
	d0	78	25	5d	29	37	66	91	dd	5a	fb	67
	6b	f3	9e	43	dc	70	11	23	1c	62	49	ad
	13	e9	87	ef	1b	8f	75	32	bd	21	db	bf
	ff	30	a5	c7	5e	92	2a	7f	64	65	46	a7
	9e	4	e8	8d	4e	64	9b	81	35	70	4	68
	7b	ca	8f	97	3d	a5	ae	6a	b4	88	25	f9
	64	5	1c	c7	99	2e	6d	9f	2c	e6	73	9a
	89	cd	8d	50	3d	46	6e	3d	80	c9	d1	a1
	52	7a	86	7e	b0	76	35	d	9f	c	80	f
	a3	d8	f0	cc	f9	e7	4e	81	2f	ae	ca	e7
	88	7c	12	7e	26	d6	98	56	c	bd	a9	db
	8e	bd	e8	a0	94	5a	62	f3	aa	e4	48	d1
	e6	d1	56	18	2c	40	40	e6	ce	62	fc	d2
	3c	a4	3a	56	27	d1	fe	ae	de	0	db	d4
	3d	7b	b6	66	f	ac	b9	ee	67	36	7c	e0
	fb	b3	f	4	6c	84	c1	61	d9	84	5b	2d
	65	19	14	a1	a9	cf	69	fc	10	a7	6b	11
	41	bf	33	6c	98	63	0	9a	45	45	c3	a5
	76	88	7	a2	f1	4b	8b	1d	38	8f	5c	75
	c8	12	60	df	f4	16	0	e1	b2	43	b6	9f
	a1	fb	1c	8c	14	d8	86	a1	75	34	8c	38
	53	4f	b1	7d	b6	4c	2e	ef	5b	8b	f4	3d
	58	85	60	ce	45	d9	9d	70	63	b8	a1	c8
	f	14	a7	de	ea	69	50	55	1	a8	27	e8
	3a	a2	25	7a	7e	9d	cf	f8	c9	1a	d7	86
	b6	69	6c	df	3c	68	35	87	2e	93	f1	d7
	cd	69	e6	62	ad	57	1a	23	1f	4b	39	83
	ce	48	8	90	ca	57	c7	96	d2	3a	c1	95
	a6	6c	15	67	f1	63	ba	2f	5d	d0	c1	
d0	8a	d7	57	d3	c8	b	17	32	59	8e	56	2
	41	f1	f7	57	51	d3	61	dc	58	2f	99	3a
	b7	d7	f1	ab	9b	78	e0	bd	ef	5	4c	f8

	54	54	b0	e4	19	2e	3a	ed	92	d5	e4	6e
	9	92	aa	23	e6	6a	1d	9d	25	3a	ee	7c
	a5	df	7f	63	b2	bc	9	88	46	3e	40	4c
	5e	aa	27	8c	b	9f	d2	7b	7c	8	10	b9
	3c	2b	9c	ec	ac	cc	91	d6	6f	8	13	6c
	a	4	5e	99	4	a8	a0	49	d7	d8	d2	18
	3b	98	40	d4	1f	94	36	56	b8	35	48	78
	10	2d	ca	82	d5	da	ff	58	fe	a5	b7	a1
	3f	9f	91	ae	b3	a2	81	c9	ba	5a	d2	f
	d5	91	da	8d	8d	f8	35	d2	ce	ea	c0	d1
	8a	1c	5d	ab	8b	1a	84	11	f	e4	a2	26
	77	34	ef	30	77	7c	86	40	b5	73	e8	39
	b4	c1	e6	f1	99	ac	d4	27	68	ac	37	7d
	54	70	bb	ed	27	db	44	71	1c	a6	bc	15
	99	d	cd	ad	a4	15	7a	70	42	aa	d	27
	ec	2d	7f	7d	fa	28	e7	6b	6c	42	32	b0
	98	a5	e2	10	b0	1e	d7	52	ae	2f	f8	14
	68	a8	f5	d4	52	6a	52	d2	a8	97	a5	1c
	c9	9f	36	b9	90	47	b7	5d	5e	a0	45	1
	ce	c4	6e	ac	e	6	b5	c9	27	32	10	6f
	f4	37	bb	f7	a8	a5	94	d5	bf	7c	49	f2
	47	79	d3	55	fe	1	6e	28	49	98	1c	fa
	5e	c	eb	6d	cc	44	17	22	53	5c	d	52
	15	8	d2	97	1d	27	32	dd	a5	2f	24	72
	c7	85	d	7d	b9	64	2d	7d	e5	b7	41	5
	38	fc	62	a	bb	a6	b5	d8	f2	76	b0	46
	25	56	42	32	24	78	d0	5d	d5	ce	a6	4
	cf	59	9b	8f	42	2	88	a	c7	2a	e8	7
	f8	34	8b	38	b5	a7	c4	14	43	cc	62	
6	8f	a2	6f	ed	f4	4c	ec	d	a1	8f	b5	a4
	e4	12	bc	38	51	0	a2	66	4f	21	c	31
	6e	7a	1a	23	99	c	7	45	e8	7c	6c	5a
	20	7c	b0	2	fe	be	3	c8	f3	c2	e3	9b
	f0	64	d6	2d	d8	e8	fa	63	36	af	78	74
	5	7	35	6e	8c	cc	92	7a	8d	2b	f3	e9
	b8	95	85	68	28	fa	b7	62	ff	d	90	99
	10	3d	61	28	43	de	e6	1b	48	27	12	ce
	32	ef	11	e7	a6	27	30	51	9f	61	73	41
	a0	5c	f0	a8	62	db	1e	d2	3d	74	c1	78
	76	d9	12	11	84	8a	d1	8c	97	f0	47	d3

	52	c2	98	e9	67	a4	28	c7	5b	6a	4e	8
	f2	90	b6	e	1d	a4	b4	87	d9	cf	43	56
	61	c9	3b	f4	50	33	81	35	a7	3a	46	f1
	dd	e8	7f	49	59	8	cb	8b	d7	fd	f2	75
	7c	60	4a	a0	e5	75	fb	b6	da	f2	8c	3a
	7c	93	63	55	7d	cf	e2	4a	5c	26	a7	e4
	76	f1	8f	6c	f	9d	12	77	a0	2f	20	95
	22	92	8d	f9	b1	d6	e7	10	c2	fb	8f	5e
	6d	e	b3	d0	a8	79	6d	2d	5e	a2	cc	d8
	eb	74	14	f9	5f	b8	8c	53	b1	ef	8	6f
	cd	82	93	83	6d	86	27	72	3a	69	56	af
	bd	0	f9	23	1	c4	c0	18	59	b7	6d	71
	5c	b9	a0	c9	f0	9e	7	e7	d	38	ae	54
	ef	3d	e	a7	64	dd	80	75	bf	66	ed	7f
	6c	f9	3a	a3	49	29	d4	e6	42	63	cf	40
	80	a6	f6	48	5c	66	db	96	c4	c7	2	9d
	8e	6a	50	15	42	fb	5f	ba	3b	e3	94	aa
	be	87	78	e8	b0	57	8d	9a	40	30	4e	9
	e2	19	2b	2c	12	96	5b	d8	9b	91	c6	3a
	c3	d8	dc	2b	a5	8c	8c	3	dc	fc	7b	7
	81	60	71	fb	32	bd	47	6f	71	a5		
b3	e6	bc	f8	2	32	30	dc	93	90	e9	f2	b0
	c9	88	2	a7	ad	b0	c9	62	c3	68	b9	da
	e1	90	f3	f4	22	bf	4c	df	db	d4	d3	2e
	8	fd	2b	f	f4	fe	69	60	90	2b	d5	ec
	83	88	3b	a8	72	3a	b8	32	e8	62	2e	95
	6d	49	6f	92	3e	df	4a	84	4e	c9	86	6
	e0	95	82	5a	10	19	3c	4c	a0	1d	e0	25
	7d	a5	f0	83	7e	c7	a6	33	6b	5a	31	b3
	7e	98	4e	74	a6	2e	b4	3e	92	a7	7b	7b
	a6	f1	41	ae	cf	a	c6	33	c8	f4	82	6c
	d6	ed	8f	6d	e5	83	6a	7e	95	67	a2	64
	f7	1f	db	6d	cd	c7	c4	a4	1	54	1c	96
	54	a	60	63	5c	b7	30	f9	27	8e	ec	95
	cc	7e	ea	db	38	26	d7	a3	46	b6	f0	8c
	5f	e0	37	60	35	6a	9a	de	e6	b1	dc	cc
	80	8c	c9	f0	d8	fd	3d	85	49	3a	c8	64
	a3	8	44	43	a6	f5	62	21	81	cd	d4	7f
	3	61	b0	f6	ad	bc	6d	69	79	f0	39	26
	9c	ad	96	9a	ac	58	26	25	68	5b	d7	2f

7c	f3	d1	bf	f	59	13	16	4d	ea	31	38
b9	b2	ba	bc	ab	4	3d	e	b4	73	db	70
30	53	67	cf	2d	59	cc	c5	68	41	b9	90
bb	2e	a2	bc	cb	49	c9	fb	5e	79	1b	8b
66	f4	fa	65	c3	62	48	a	97	b6	4f	ae
8c	98	91	28	76	33	23	43	8c	44	d	1d
a4	51	23	35	8d	7f	12	a7	59	35	30	4e
3c	de	f	5	a8	d6	2f	81	65	3d	26	d
a2	9	bd	ba	c8	80	15	14	a9	5e	8b	9a
39	8f	2b	fd	ae	86	3b	72	91	87	f0	12
d8	27	8a	30	82	c	fa	1a	3f	b9	57	dd
57	98	17	f7	49	9b	86	ef	b5	15	5f	6f
f	fb	ca	94	7a	7d	e7	cc	5e	79	2	

Результат расшифровки:

Он, однако ж, не то чтоб уж был совсем в беспамятстве во всё время болезни: это было лихорадочное состояние, с бредом и полусознанием. Многое он потом припомнил. То казалось ему, что около него собирается много народу и хотят его взять и куда-то вынести, очень об нем спорят и ссорятся. То вдруг он один в комнате, все ушли и боятся его, и только изредка чуть-чуть отворяют дверь посмотреть на него, грозят ему, сговариваются об чем-то промеж себя, смеются и дразнят его. Настасью он часто помнил подле себя; различал и еще одного человека, очень будто бы ему знакомого, но кого именно - никак не мог догадаться и тосковал об этом, даже и плакал. Иной раз казалось ему, что он уже с месяц лежит; в другой раз - что всё тот же день идет. Но об том - об том он совершенно забыл; зато ежеминутно помнил, что об чем-то забыл, чего нельзя забывать, - терзался, мучился, припоминая, стонал, впадал в бешенство или в ужасный, невыносимый страх. Тогда он порывался с места, хотел бежать, но всегда кто-нибудь его останавливал силой, и он опять впадал в бессилие и беспамятство. Наконец он совсем пришел в себя.

Произошло это утром, в десять часов. В этот час утра, в ясные дни, солнце всегда длинною полосой проходило по его правой стене и освещало угол подле двери. У постели его стояла Настасья и еще один человек, очень любопытно его разглядывавший и совершенно ему незнакомый. Это был молодой парень в кафтане, с бородкой, и с виду походил на артельщика. Из полуотворенной двери выглядывала хозяйка. Раскольников приподнялся.